AN INTEGRATED ARCHITECTURE AND FEATURE SELECTION ALGORITHM FOR RADIAL BASIS NEURAL NETWORKS

THESIS

Timothy D. Flietstra, Captain, USAF

AFIT/GOR/ENS/02-07

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U. S. Government.

AFIT/GOR/ENS/02-07

AN INTEGRATED ARCHITECTURE AND FEATURE SELECTION ALGORITHM FOR RADIAL BASIS NEURAL NETWORKS

THESIS

Presented to the Faculty

Department of Operational Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Operations Research

Timothy D. Flietstra, BS

Captain, USAF

March 2002

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT/GOR/ENS/02-07

AN INTEGRATED ARCHITECTURE AND FEATURE SELECTION ALGORITHM FOR RADIAL BASIS NEURAL NETWORKS

Timothy D. Flietstra, BS
Captain, USAF

Approved:

/s/_____ _____
Kenneth W. Bauer, Ph.D. (Advisor) date

/s/_____ _____
Jeffrey P. Kharoufeh, Ph.D. (Reader) date

Acknowledgments

I would like to express my sincere appreciation to my faculty advisor, Dr. Kenneth Bauer, for his guidance and support throughout the course of this thesis effort. The insight and direction was certainly appreciated. He also gave me the latitude and freedom to explore the problems on my own. I would also like to thank my reader, Dr. Jeffrey Kharoufeh, for the direction he provided in the preparation of the thesis. His input has been invaluable.

I wish to also thank my family and friends for their invaluable support, patience and understanding through this process. Without them, the completion of this project would not have been possible.

Timothy D. Flietstra

Table of Contents

List of Figures

List of Tables

Page

List of Abbreviations and Acronyms

AER	Actual Error Rate
ANN	Artificial Neural Network
CM	Confusion Matrix
DA	Discriminant Analysis
DBS	Derivative Based Saliency
FFNN	Feed-Forward Neural Network
FN	False Negative
FP	False Positive
GRNN	General Regression Neural Network
$\mathbf{P_D}$	Probability of Detection
$\mathbf{P_{FA}}$	Probability of False Alarm
PDF	Probability Density Function
PNN	Probabilistic Neural Network
RBNN	Radial Basis Neural Network
RICA	Radial Basis Function Iterative Construction Algorithm
ROC	Receiver Operating Characteristic
SNR	Signal-to-Noise Ratio
$\mathbf{SNR^{RBNN}}$	Signal-to-Noise Ratio clustering for Radial Basis Neural Networks
TN	True Negative
TP	True Positive
UWBCD	University of Wisconsin Breast Cancer Data

Abstract

The research contribution of this thesis is the first known integrated architecture and feature selection algorithm for Radial Basis Neural Networks (RBNN's). The objective is to apply the network iteratively to determine the final architecture and feature set used to evaluate a problem. Additionally, this thesis compares three different classification techniques, Discriminant Analysis (DA), Feed-Forward Neural Networks (FFN) and RBNN's against several hard to solve problems. These problems were used to evaluate general classifier performance as well as the performance of the feature selection techniques.

This thesis describes the classification techniques as well as the measures used to evaluate them. It next develops a new clustering technique used to determine the network architecture and the saliency measure used to select features for RBNN's. Next, the thesis applies these techniques to three general problems, Block-C, the University of Wisconsin Breast Cancer Data (UWBCD) and a noise corrupted version of Fisher's Iris problem. Finally, the conclusions and recommendations for future research are provided.

AN INTEGRATED ARCHITECTURE AND FEATURE SELECTION ALGORITHM FOR RADIAL BASIS NEURAL NETWORKS

1 Introduction

1.1 General Discussion

The science of classification deals with a general class of problems wherein real-world observations are used to distinguish between two or more classes of interest. One example of classification is a college admissions department attempting to distinguish individuals who will graduate from those who will not. Another example is the classification of certain cells as cancerous or benign. Military applications include automated classification of images as target or clutter. There are numerous approaches to classification, encompassing qualitative and quantitative techniques. The focus of this thesis is on quantitative techniques including discriminant analysis (DA) and artificial neural networks (ANN).

Regardless of the approach used, there will likely be errors in determining the class in which an observation belongs. Associated with misclassification errors are costs or losses. Some costs are minimal, such as denying college admission to someone who would graduate. This will only hurt an institution if they do not admit and graduate enough students to make money. In other situations however, misclassifications can have very serious consequences. If cancerous cells are misdiagnosed as benign, lives could be lost. The goal of all classifying problems is to minimize misclassifications, particularly

those that are very costly. Therefore, it is important to understand the situations where classifiers will perform well, as well as the situations where they struggle.

There are certain problems for which some classifiers perform poorly. Alsing [1], in evaluating competing classifiers, presented several challenges to a linear or quadratic discriminant classifier. Data that is not separable in a linear or quadratic fashion defeats linear and quadratic classifiers. Examples of such problems include XOR data, the Block C problem (Figure 1-1) and the Iron Cross problem (Figure 1-2.) These problems depart from multivariate normality into the realm of pattern recognition as it might be applied to image classification and human behavior.

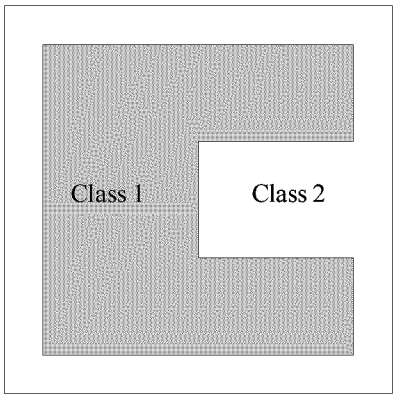

Figure 1-1. Block C Problem

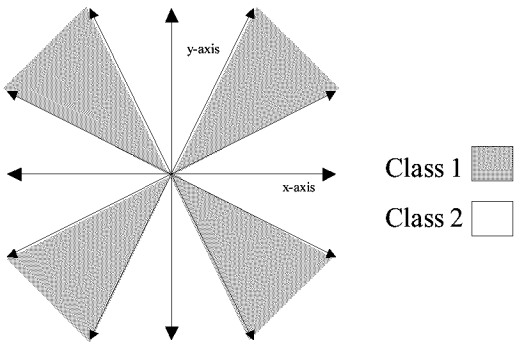

Figure 1-2. Iron Cross

The dimensionality of the data can also pose problems for a classifier. G.V. Trunk [17] purports that prediction accuracy of a classifier will drop to 50% as the number of dimensions in the data increases for a finite data set. In his application, he adds real features to the exemplars, with the distance between the two classes for each successive feature approaching zero. Classification is accomplished using a simplified classifier, which assumes the distribution of the two classes, and does not estimate this information from the data. While these assumptions are not viable for the techniques that will be discussed in this thesis, it does suggest that the number of features has a detrimental impact on classification accuracy. This thesis will explore the relationship between dimensionality and classification accuracy for DA and ANNs. It will also measure the impact that feature selection, the removal of insignificant features, has on classifier performance.

DA and ANNs are generally used for classification and pattern recognition problems [20]. These classifiers attempt to map the input vectors to vectors of ones and zeros (depending on the number of classes in the problem). In addition to classification problems, ANNs can be applied to nonlinear regression [20]. Radial basis neural networks (RBNN) can be employed in a generalized regression neural network (GRNN) framework. In this framework the networks fit a nonlinear function to the input data, providing a function as output instead of a classification vector or value [19]. A special case of nonlinear regression is time series analysis, where the features are the previous responses (in time) with some delay [9].

1.2 Problem Statement and Research Objectives

This thesis will compare the efficacy of the aforementioned classifiers using several techniques explored in Alsing [1]. One measure used will be classification accuracy – an estimate of the Actual Error Rate (AER) calculated from applying the

classifier developed against an independent validation data set. Receiver Operating Characteristic (ROC) curves will also be used to compare the impact of differing decision criteria on Type I and II errors. Lastly, a Multinomial Selection procedure will be used to rank the classifiers over the different problems.

Hard-to-solve problems will be explored in relation to the classifiers. The problems evaluated will include general classification and feature selection problems. This thesis will explore the problems dimensionality poses to a general classification problem. It will also analyze different pattern recognition problems of varying complexity to challenge the classifiers. Finally, it will apply the classification techniques against breast cancer data from the University of Wisconsin [18] and Fisher's Iris Problem [4].

The goal of this research is two-fold. The main research objective is to develop an integrated architecture and feature selection algorithm for RBNN's. This feature selection algorithm will be compared with the feature selection techniques for the other classifiers. A secondary goal included in this effort is to evaluate the overall effect feature selection has on classification accuracy across the classifiers.

Further, different classifiers will be evaluated against a set of challenging problems. The goal is to explore differences in classifier performance against a broad set of problems and to develop a methodology to determine the appropriateness of different classification techniques for these problems. This will aid in determining the best alternatives for different problem types.

2 Literature Review

2.1 Overview

This chapter reviews the literature regarding the classifiers under discussion and various evaluation criteria used for classifiers. The research is focused on the area of feature selection. For Discriminant Analysis (DA), there is a discussion of two approaches to feature selection: Stepwise DA and Discriminant Loadings (DL). The literature review regarding Feed Forward Neural Networks (FFNN) will cover network architecture, backpropagation and feature selection. For the last classifier, Radial Basis Function Neural Networks (RBNN), there is no developed feature selection algorithm; several proposed solutions will be explored in chapter 3. The literature review for RBNN will concentrate on network architectures, kernel functions and clustering algorithms.

2.2 Discriminant Analysis (DA)

DA classifies exemplars into groups by creating a hyperplane – either linear or hyperbaloid – to separate the feature space into two distinct areas (for the two-group problem). This decision line is based on the within-class mean vectors and the covariance structure of the features. If the two classes are linearly or quadratically separable, DA will perfectly differentiate between the two classes if the appropriate form is used.

A key assumption for DA is that the independent variables must possess a multivariate normal distribution [6]. While the technique remains robust against small departures from normality, if the data severely departs from this assumption, classification accuracy can be greatly affected. Additionally, this can impact the statistical method of feature selection, Stepwise DA, discussed below.

The second assumption impacts the DA method used – Fisher's Approach or Quadratic Discrimination. To use Fisher's Approach, the within class covariance structure must be equal for the two groups being classified. This assumption can be tested using the following hypothesis test [3]. The null hypothesis states that the within class covariance matrices are from the same underlying distribution. Under the null hypothesis

$$P\{-2\rho \ln W_1 \le Z\} = P\{\chi^2_F \le Z\} \tag{2.1}$$

where q = number of groups, p = number of variables, N = total sample size, $n = N - q$, N_g = number in group g, $n_g = N_g - 1$ and F the degrees of freedom for the test, and where,

$$\rho = 1 - \left(\sum_{g=1}^{q} \frac{1}{n_g} - \frac{1}{n} \right) \left(\frac{2p^2 + 3p - 1}{6(p+1)(q-1)} \right) \tag{2.2}$$

$$\ln W_1 = \sum_{g=1}^{q} \frac{1}{2} n_g \ln \left| \Sigma_g \right| - \frac{1}{2} n \ln \left| \Sigma \right| \tag{2.3}$$

$$F = \frac{1}{2}(q-1)p(p+1) \tag{2.4}$$

If the test statistic, $-2\rho ln W_1$, is sufficiently large, we reject the null hypothesis and conclude the within class covariance structures are unequal.

2.2.1 Fisher's Approach.

Under the assumption of a common covariance structure, Fisher's approach can be applied to solve the problem. Fisher sought to maximize the following equation

$$\frac{\left(b^T \mu_1 - b^T \mu_2 \right)^2}{b^T \Sigma b} \tag{2.5}$$

This equation describes the squared distance between the discriminant scores of the two class means ($b^T \mu_i$) with respect to the variance of the discriminant scores ($b^T \Sigma b$) [3]. The solution $\underline{b}$ to solve this nonlinear program is [6]

$$\underline{b} = \Sigma^{-1}(\mu_1 - \mu_2) \tag{2.6}$$

For any practical problem, the true population parameters are unknown, and therefore, need to be approximated using the sample means and covariance as unbiased estimators of the true parameters.

To classify a new exemplar, the linear combination is applied to the new data point. In this thesis, the prior probabilities of the two groups are assumed to be equal, as well as the "costs" of misclassification. In this problem, exemplars are classified according to which side they are of the midpoint of the centroids (mean vectors) in projected space which is

$$M = \frac{\overline{Y}_1 + \overline{Y}_2}{2} = \frac{1}{2}\left(\overline{X}_1 - \overline{X}_2\right)^T S^{-1}\left(\overline{X}_1 + \overline{X}_2\right) \tag{2.7}$$

The decision rule (in projected space) becomes: If $Y_{new} = \underline{b}^T X_{new} > M$, classify as Group 1 – otherwise classify as group 2. This assumes the projection of the group one centroid is larger in the projected space than that of the second group.

2.2.2 Quadratic Discriminant Functions

The quadratic discrimination approach provides a greater ability to separate classes – particularly if the classes are not linearly separable. This approach is necessary if the covariance structure is different for the two classes, and allowing for these differences provides the greater flexibility. This approach is also easily extended to more than two classes. Each class generates its own quadratic discriminant score [6]

$$d_{Q_i} = -\frac{1}{2}\ln\left|\Sigma_i\right| - \frac{1}{2}\left(\underline{x} - \mu_i\right)^T \Sigma_i^{-1}\left(\underline{x} - \mu_i\right) + \ln\left(P_i\right) \tag{2.8}$$

where P_i is the prior probability of the exemplar belong to class *i*. The decision rule is very simple; an exemplar is classified according to the largest discriminant score. This approach will produce results identical to Fisher's equation if the within-class covariance matrices are identical. Because of the flexibility, greater classification power provided by

this approach and the relaxation of the assumption of equal within-class covariance structures (although multivariate normality is now assumed), quadratic discriminant functions will be used for all applications discussed in this thesis.

2.2.3 Feature Selection

As discussed previously, two different approaches to feature selection will be explored, Stepwise DA and Discriminant Loadings. Both applications will be discussed in a backward selection paradigm – all the features will be included, and one feature will be removed at a time according to a selection criteria.

Stepwise DA employs partial *F*-tests similar to stepwise regression. Without multivariate normality, the *F* statistics will not accurately describe the significance of the individual features. If the data is taken from a multivariate normal distribution, the following statistic is distributed as $F_{(p-1,\, N-p-1)}$ [8]

$$F = \left(\frac{N-p-1}{p-1}\right)\left(\frac{N_1 N_2}{N(N-2)}\right)\left(\frac{\Delta^2{}_p - \Delta^2{}_{p-1}}{1+\left(\frac{N_1 N_2}{N(N-2)}\right)\Delta^2{}_{p-1}}\right) \tag{2.9}$$

where N = total sample size, p = number of variables, N_i = number in group i, and Δ^2_i are the Mahalanobis distance between the respective group means, defined to be [6]

$$\Delta^2{}_i = (\mu_1 - \mu_2)^T \Sigma^{-1} (\mu_1 - \mu_2) \tag{2.10}$$

This test statistic compares the distance between the means with all p features, Δ^2_p, with the Mahalanobis distance with one feature removed, Δ^2_{p-1}. A feature is considered significant if $F > F_\alpha$, the null hypothesis being that the feature is not significant. Under a backward selection routine all features are included in the original model. During each iteration, the *F* statistic is calculated for each feature, and the least significant feature is

removed (the feature with the smallest F value) [8]. This process continues until all the insignificant features are removed or until only the most significant feature remains.

Discriminant Loadings provide an alternative to Stepwise DA, and do not require the assumption of multivariate normality; however, the technique does assume equal within-class covariance structures. Discriminant Loadings provide the correlation of a feature with the discriminant function. Loadings have the following form [3]

$$DL = RD_{\tilde{X}}^{\frac{1}{2}} \underline{b}\left(\underline{b}^T C \underline{b}\right)^{-\frac{1}{2}} \tag{2.11}$$

where C is the sample covariance of X, $D_{\tilde{X}}$ is the matrix of the diagonal elements of C and R is the sample correlation of X. It is assumed that the least significant feature has the smallest loading in absolute value. Similarly, the most significant feature has the largest loading. As with Stepwise DA, Discriminant Loadings can be applied in an iterative manner. For each iteration, the loadings are calculated and the feature corresponding to the smallest loading is removed.

Dillon and Goldstein [6] assert that Discriminant Loadings provide a clearer indication of which features are important. The loadings reflect common variance among the predictors, and are less subject to multicollinearity among the features. The partial F-values used in Stepwise DA however, can be confounded by highly correlated features. For these reasons, this thesis will employ Discriminant Loadings to perform feature selection.

2.3 Feed-Forward Neural Networks

FFNN's (as well as the other Artificial Neural Networks (ANN)) employ a completely different approach to classification than DA. ANN's are loosely based on a biological concept. Neurodes are connected and information is passed between them. The key to using this structure for classification is the updating of the information being

passed. In FFNN's, this process is called learning, and its goal is to produce outputs that closely resemble the class membership [3]. Figure 2-1 illustrates a standard FFNN. There are generally three layers to the network: Input, Hidden and Output. The upper layers receive a weighted sum of the outputs of the previous layer's nodes. Inside the node, a threshold function is applied to this sum, restricting the function values to the interval [0,1] or [-1,1]. The most commonly used threshold function is the sigmoid function (see Figure 2-2). It restricts the network output to the interval [0,1], and most importantly is differentiable. This is critical for backpropagation to work. It has the following form

$$f(a) = \frac{1}{\left(1 + e^{-a}\right)} \tag{2.12}$$

With enough nodes in the hidden layer, FFNN are universal function approximators. A FFNN is an ANN where all the connections move from lower to higher levels.

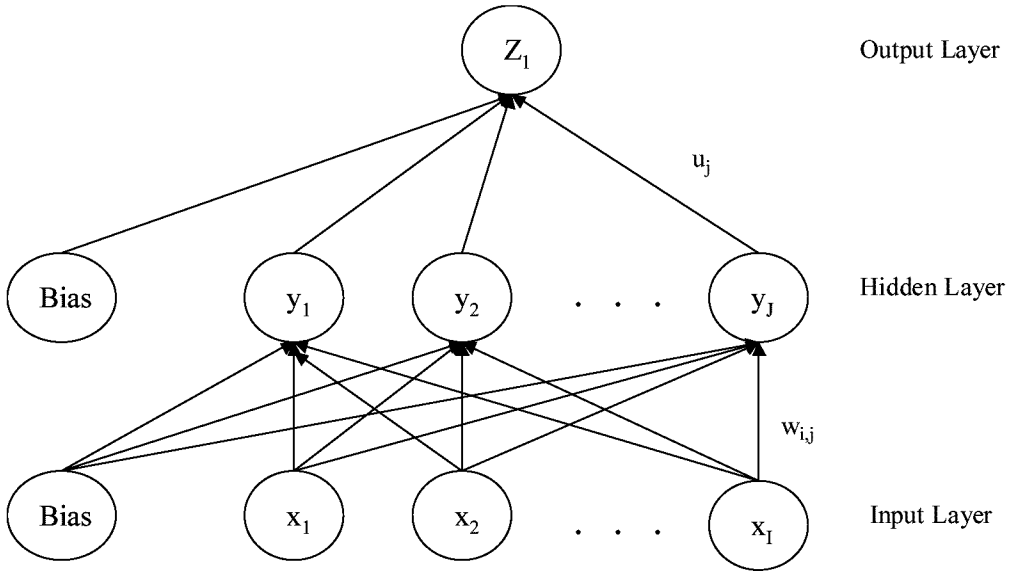

Figure 2-1. FFNN with Bias and Single Output [3]

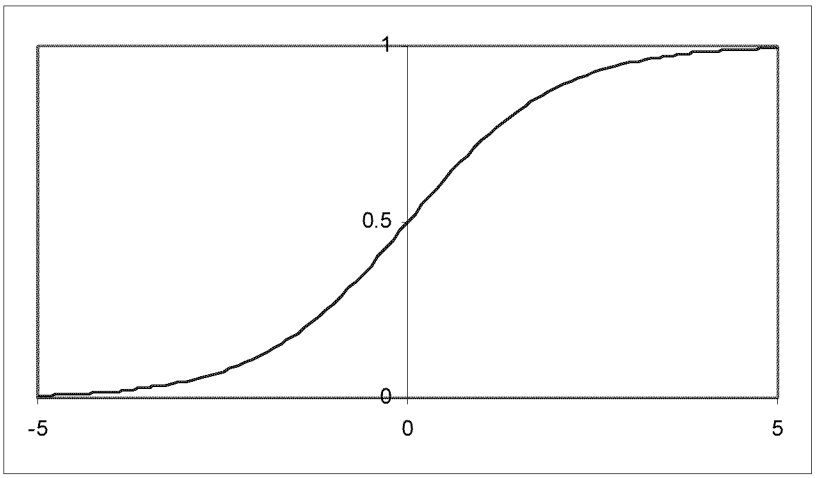

Figure 2-2. Sigmoid Function

2.3.1 Backpropagation

Backpropagation is the standard manner by which the weights are updated in a FFNN [11]. Typically, the goal of the network is to produce outputs that are very close to one for class one and zero for class two. The weights are adjusted during training to minimize the total squared error

$$E = \sum_{i=1}^{n} \left(t^{(i)} - z^{(i)} \right)^2 \tag{2.13}$$

where n is the number of exemplars, $t^{(i)}$ is the target and $z^{(i)}$ is the network output for the i^{th} exemplar. The weights are initialized randomly, and then a gradient descent routine is used to iteratively update the weights. The weights are updated until the error converges, or until we have cycled through the data (an epoch) the maximum number of times. For each exemplar, the error is calculated. The weights are updated according to the gradient of the error with respect to the weights. First the upper weights, u_k (see Figure 2-1), are updated, and then are used to update the lower weights, $w_{j,k}$. The weight updates for the upper weights for the i^{th} exemplar have the following form

$$u_k^{(r+1)} = u_k^{(r)} + \eta\left\{\left(t^{(i)} - z^{(i)}\right)\left[z^{(i)}\left(1 - z^{(i)}\right)\right]y_k^{(i)}\right\} \tag{2.14}$$

where $y_k^{(i)}$ is the output of the k^{th} hidden node for exemplar i and η is the learning rate (preferably around 0.01). The lower weights are updated in the following fashion

$$w_{j,k}^{(r+1)} = w_{j,k}^{(r)} + \eta\left\{\left(t^{(i)} - z^{(i)}\right)\left[z^{(i)}\left(1 - z^{(i)}\right)\right]y_k^{(i)}\right\}\left[y_k^{(i)}\left(1 - y_k^{(i)}\right)\right]x_j^{(i)} \tag{2.15}$$

where $x_j^{(i)}$ is the j^{th} feature of the i^{th} exemplar.

Apart from a strict gradient search routine, there are many techniques that are used to accelerate convergence [12]. These techniques include the Conjugate Gradient Method, which uses a second-order approximation of the gradient along which to move. Momentum modifies the gradient by adding a first-order term containing the previous weight update, and is used to smooth the direction of descent. Adaptive learning adjusts the learning rate around a minima, by shrinking the step size. This thesis will employ MATLAB®'s "traingdx" routine, with a momentum coefficient of 0.9, and adaptive learning rates of 1.05 and 0.7 for increasing and decreasing the learning rate respectively.

2.3.2 Feature Selection

There are two main forms of feature selection for FFNN, derivative-based and weight-based saliency [3]. Derivative based saliency techniques measure the change in unit output per unit change in each of the features. For FFNN's, this is generally approximated and not calculated in closed form. Weight-based saliency instead uses the lower layer of weights to determine feature significance. The saliency measure for feature i is

$$\tau_i = \sum_{j=1}^{J} w_{i,j}^{\ 2} \tag{2.16}$$

where J is the number of hidden nodes. The smaller the saliency measure, the less significant the feature.

While both saliency measures provide a numerical scale for feature significance, neither measure provides a criteria for what is truly significant. Bauer et. al. [4] have proposed an objective criteria for determining significance, the Signal-to-Noise Ratio (SNR) Saliency Measure. In this technique, a noise feature is added to the data prior to training, taken from a Uniform(0,1) population for both classes. After training is accomplished, the weights for this feature should remain close to zero. The other feature's weight-based saliency measures are then compared to the noise variables saliency, and the SNR for feature *i* becomes

$$SNR_i = 10\log_{10}\frac{\tau_i}{\tau_N} \tag{2.17}$$

where τ_N is the saliency for the noise variable. Those features with a SNR less than zero are determined to be insignificant, and can be removed from the data set. Some care must be taken in removing features, since the initial weights can greatly impact this measure. Training several networks with different random weights can provide more confidence in the significance of different features.

2.4 Radial Basis Function Neural Networks (RBNN)

RBNN differ from FFNN in several very fundamental ways. Both general network architecture and training differ between the two. RBNNs belong to the general class of probabilistic neural networks (PNN). Under the PNN paradigm, classification is performed by estimating a probability density function (PDF) for each class. A new exemplar is classified according to the class whose density function is more likely. Unlike FFNN's, PNN's do not require training. A training set is read in, and is used to generate the PDF's for each class [19].

Kernel density estimation is the process by which the PDF's are estimated. A kernel density function is any function *K* satisfying the following equation [15]

$$\int_{-\infty}^{\infty} K(x)dx = 1 \tag{2.18}$$

Kernels are typically symmetric, though not necessarily. The Epanechnikov kernel is the most efficient kernel density function; the kernel minimizes the integrated square error of the estimator. It has the multivariate form

$$K_e(x) = \begin{cases} \frac{1}{2c_d}(d+2)(1-x^T x) & x^T x < 1 \\ 0 & otherwise \end{cases} \tag{2.19}$$

where c_d is the volume of the d-dimensional unit sphere [15]. Figure 2-3 illustrates the univariate form of the Epanechnikov.

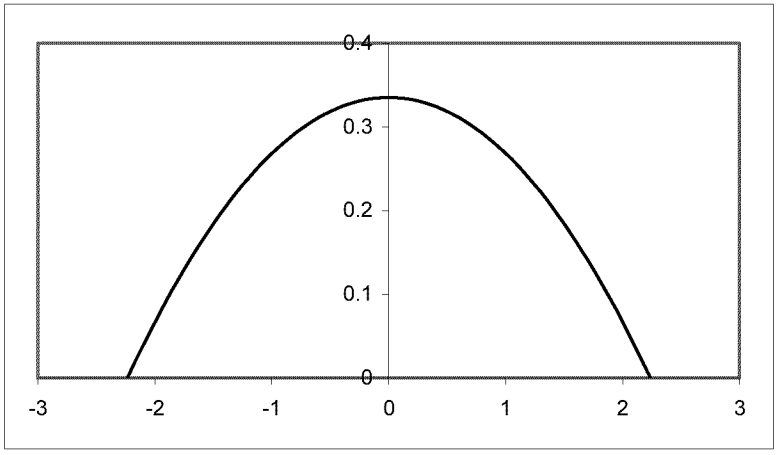

Figure 2-3 Univariate Epanechnikov Kernel

Although the Epanechnikov kernel is the most efficient method, the choice of kernel functions is relatively insignificant. Efficiency of every other kernel estimator is compared as a ratio to the Epanechnikov kernel. For example, the Gaussian kernel is approximately 95% efficient, and is the most widely used kernel estimator, particularly for PNN [19]. The Gaussian kernel has the multivariate form

$$K(x) = \frac{1}{\sqrt{(2\pi)^d}} e^{\left(-\frac{1}{2}x^T x\right)} \tag{2.20}$$

The PDF is the sum of the kernels, with each weighted by *1/N*, keeping the resulting function a PDF (maintaining the property of equation 2.18) [15].

Under the PNN paradigm, each basis function output is weighted equally. RBNNs allow the weighting for each output to be different. For RBNN, the hidden layer is made up of kernel functions centered at each exemplar of the training set (in its simplest form). Each exemplar in whole is passed to each neurode, where the kernel function maps the *n*-dimensional input vector into the real numbers. This leads to the general network architecture seen in Figure 2-4.

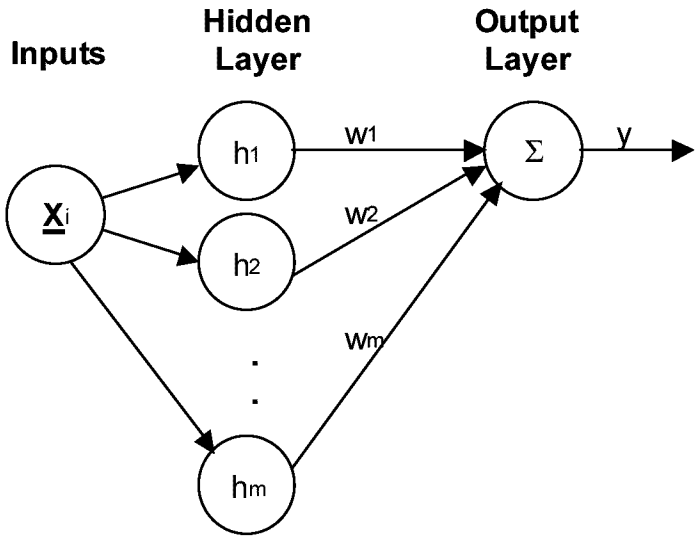

Figure 2-4. RBNN with Single Output

In this thesis, the standard function in the hidden layer will be the Gaussian with the form:

$$h_i(x) = \exp\left(\frac{-(\underline{x}-\underline{\mu}_i)^T(\underline{x}-\underline{\mu}_i)}{2\sigma_i^2}\right) \tag{2.21}$$

Training is accomplished in a similar manner to backpropagation is used for FFNN [19]. As seen in Section 2.3.2, gradient search is used to find the minimum error. For RBNN,

the training algorithm is much simpler, with only one layer of weights to train. A single output network will use the following equation to update the weights

$$w_i(n+1) = w_i(n) + \eta(t - y)z_i \tag{2.22}$$

where $z_i = h_i(x)$, t is the target value, and w_i and y are as described in Figure 2.3. A single exemplar ($\underline{x}$) is passed through all the hidden neurodes to obtain the output of the network, y. Each hidden weight is then updated using Equation 2.22. When all the training exemplars are processed, one epoch is complete. This process will continue until the error is small enough.

The training for RBNN is guaranteed to converge to a global minimum if the classes are separable by hyperplanes, unlike FFNN where the training might get caught in a local minimum [11, 16]. Training for a RBNN is also considerably faster than for a FFNN. For networks of similar size, the difference in training time can be as large as three orders of magnitude [19].

Selecting the receptive fields (σ_i) for each center is also necessary. If chosen too large, the center will have too great an impact on the output of exemplars far from the center. If chosen too small, the network will only activate for those exemplars located at the centers, leaving gaps in the classifier. One method which has produced favorable results consists of setting σ_i equal to the distance between the i^{th} center and its nearest neighbor [19]. The nearest neighbor approach will be used in this thesis to estimate the receptive fields used for the radial basis functions.

2.4.1 Cluster Algorithms

Even though training is much quicker for RBNN than FFNN, subsequent application of the network to new exemplars can take much longer. The size of the network in terms of the number of hidden nodes can be much larger for a RBNN than for an equivalent FFNN [11,16]. Clustering techniques can be used to represent multiple

hidden nodes with a single node, thus reducing the computational effort required for training which is proportional to number of training vectors [12].

One must be careful not to use clustering techniques indiscriminately. As the number of features increases, clustering techniques can erroneously identify cluster centers, clustering around features which are not useful for classification [19]. This indicates that feature selection can improve clustering accuracy, which will in turn improve classification accuracy. Three clustering algorithms will be discussed next: a simplified algorithm due to Wasserman, *K*-Means and the Radial Basis Function Iterative Construction Algorithm (RICA). Supplemental flowcharts will be included for additional clarification.

Wasserman [19] presents a simple clustering algorithm, in which nodes are pruned (removed from consideration as centers) and have no impact on the centers used when the network is trained. Each class is processed, with the centers produced in a single pass through the data. The first exemplar is chosen as a basis function center. Each subsequent exemplar is processed using Euclidean distance to determine the closest center. If this distance is smaller than a threshold distance, the exemplar is discarded. If, however, the distance is larger than the threshold, the exemplar becomes a new center. One problem with this algorithm is that different sequences will produce very different results. It also discards information about the density of the training data, since nodes are pruned, instead of impacting the location of the centers.

K-Means clustering is a self-organizing procedure. Unlike the simple clustering discussed above, it is iterative, stopping when the centers selected remain the same. It derives its name from the output of the algorithm. A number of clusters (K) is specified, and the algorithm returns the means of each cluster of data [2]. Each class will be clustered separately, with K not necessarily the same for each class. There are several ways to initiate the algorithm, but the most common is to assign K random exemplars as

initial centers [6]. Each successive exemplar is assigned to the nearest center. Once all the data is assigned to a cluster, the means of each cluster become the new centers. The data is processed in the following manner until the centers remain the same between iterations [12].

Without *a priori* knowledge of the number of clusters, the selection of *K* involves experimentation. One measure for accomplishing this task is the squared sum of the deviations of each exemplar from its cluster center. Candidate *K* values for are used, and that value of *K* which produces the smallest error is selected [12]. Certain values for *K* should be excluded. If *K* is allowed to be equal to the number of exemplars, the error will be zero, and the algorithm will produce clusters equivalent to the training data. Hence, if *K* is allowed to approach the number of exemplars, too many clusters will just contain one point. For this research, *K* is limited to one half the number of exemplars for a given problem. Figure 2-5 below illustrates the algorithm in flow-chart form.

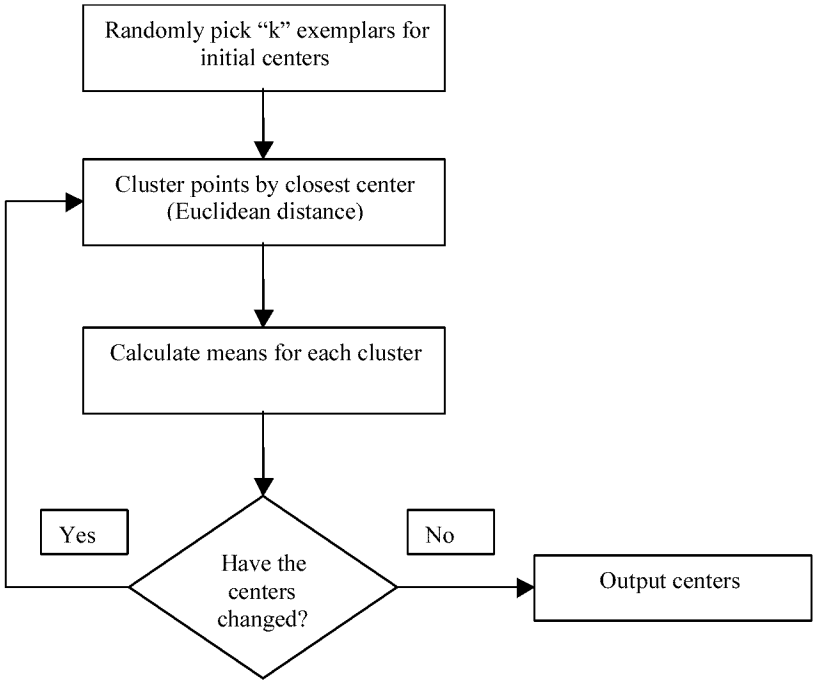

Figure 2-5. *K*-Means Algorithm

While the preceding algorithms simply define the cluster means, RICA describes the distribution of each center individually described by the mean and covariance of the cluster. The end result of the procedure is [21]

$$h_i(\underline{x}) = e^{-\frac{(\underline{x}-\mu_i)^T \Sigma_i^{-1} (\underline{x}-\mu_i)}{2}}. \tag{2.23}$$

The key to the algorithm is determining the number of clusters and their partitioning. Wilson [21] proposes using Shapiro-Wilk test statistics to determine if the current partition is sufficiently distributed as a multivariate normal. A Shapiro-Wilk test statistic for the current partition of the data is compared to the test statistic of two partitions generated from the current one. Wilson employs the univariate form of the test statistic

$$W = \frac{\left(\underline{a}_i X_{(i)}\right)^2}{Ns^2} \tag{2.24}$$

where a_i are weighting coefficients developed by Shapiro and Wilk, available in tables [5] for $n \leq 50$, $X_{(i)}$ are the ordered data and s^2 is the sample variance. For $n > 50$, Shapiro and Wilk provide the following approximations for the coefficients [14]

$$a_i = \frac{2m_i}{C} \text{ for } i \neq 1, n \tag{2.25}$$

where

$$m_i = \Phi^{-1}\left(\frac{i-0.375}{n+0.25}\right), i = 1,...n \text{ [10]} \tag{2.26}$$

with Φ^{-1} being the inverse cumulative distribution function of the standard normal distribution and

$$C = \sqrt{-2.722 + 4.083n} \tag{2.27}$$

For a_i and a_n, they propose a different approximation

$$a_n = -a_1 = \sqrt{\frac{\Gamma\left[\frac{1}{2}(n+1)\right]}{\sqrt{2\Gamma\left(\frac{1}{2}n+1\right)}}}. \tag{2.28}$$

As the data tends toward a normal distribution, the test statistic tends toward 1.0; the test statistic will approach zero for data that is distinctly non-normal [5]. If the current partition has a larger test statistic than either of the sub-partitions created, it is kept. Otherwise, the two new partitions will be kept and analyzed in the same manner [21].

The partitioning of the data is accomplished by employing *K*-Means with Mahalanobis distance used instead of Euclidean distance. Using Mahalanobis distance preserves the correlations present in the data [21]. If the data is standardized and the features are independent, the two distances will produce the same results, but this is not always the case. The original partitioning of the data is created using Euclidean distance, since there is no covariance structure for the two centers. Once the data is clustered, the sample means and covariances will be used in the next iteration. The *K*-Means algorithm is then employed iteratively as described above. Because the algorithm requires a covariance matrix for each cluster, if any partition has fewer than $p+1$ data points (p being the number of features) the algorithm will stop. If the covariance matrix does exist, its inverse will not exist if some of the features are linearly independent. This is evidenced by eigenvalues of the covariance matrix being zero. This can be rectified by replacing these eigenvalues with a threshold value of 0.5. The modified covariance matrix becomes [21]

$$C = VD^*V^T \tag{2.29}$$

where D^* is the matrix with the modified eigenvalues along the diagonal and V is the matrix of eigenvectors of the sample covariance matrix, C.

While Wilson [21] uses the univariate form of the Shapiro-Wilk test statistic, it is not clear how the multivariate data is applied. Malkovich and Afifi [13] have proposed a multivariate generalization of the test statistic

$$W^* = \frac{\left[\sum_{j=1}^{n} a_j U_{(j)}\right]^2}{\left(Y_m - \overline{Y}\right)^T A^{-1}\left(Y_m - \overline{Y}\right)} \tag{2.30}$$

where

$$A = \sum_{j=1}^{n}\left(Y_j - \overline{Y}\right)^T\left(Y_j - \overline{Y}\right) \tag{2.31}$$

and Y_m is the observation that has the maximum value over all the observations of

$$\left(Y_j - \overline{Y}\right)^T A^{-1}\left(Y_j - \overline{Y}\right) \tag{2.32}$$

The a_j are defined identically to those for the univariate test, and $U_{(j)}$ are the order statistics. The order statistics are defined by ordering the following statistics

$$U_i = \left(Y_m - \overline{Y}\right)^T A^{-1}\left(Y_i - \overline{Y}\right) \tag{2.33}$$

W^* has the same interpretation as W, namely the closer to 1, the more normal the underlying population. Using W^* instead of W in Wilson's algorithm provides a more meaningful multivariate interpretation while being computationally simpler. Figure 2-6 describes the algorithm with n denoting the number of exemplars and m the number of features.

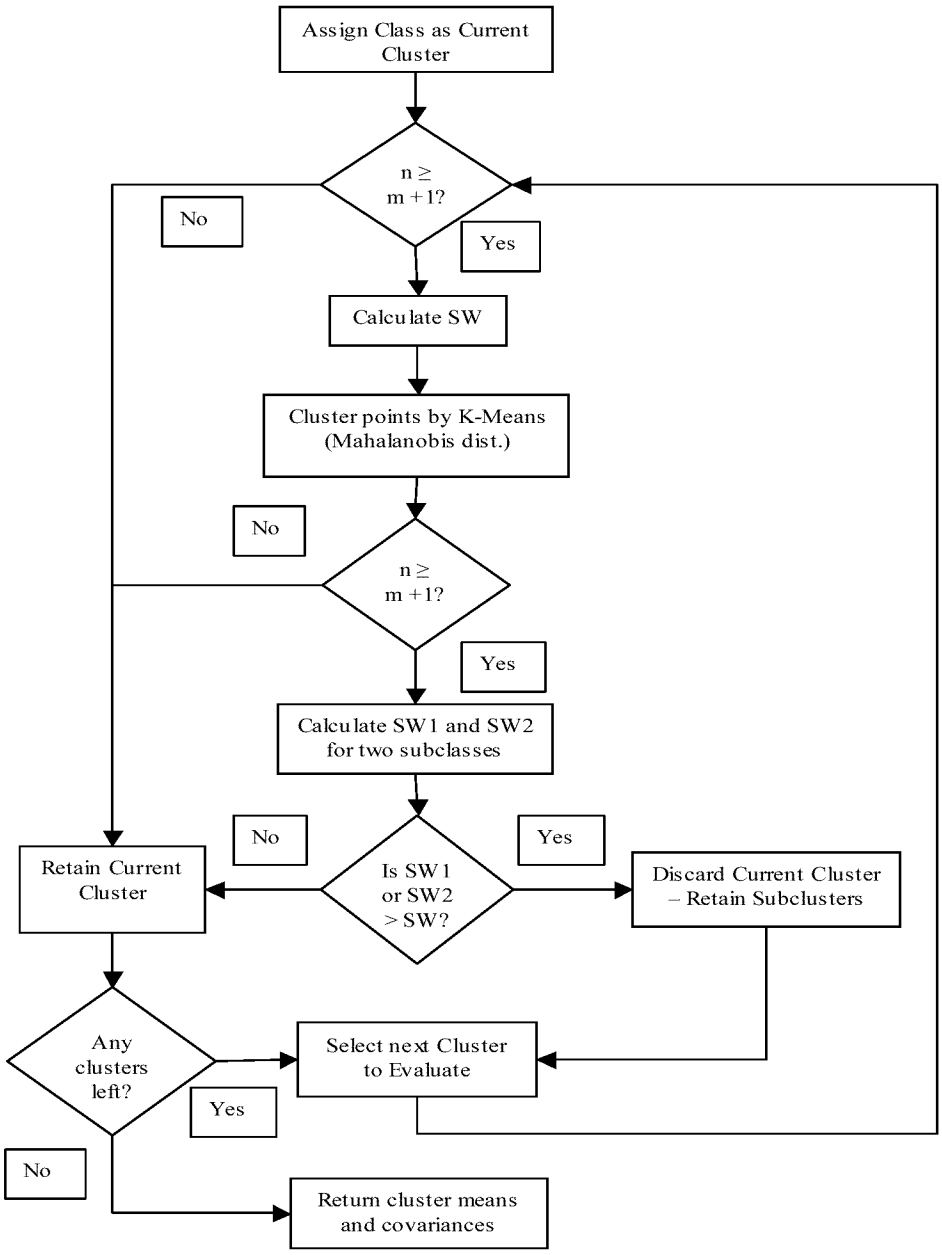

Figure 2-6. RICA Clustering Algorithm

2.4.2 General Regression Neural Network

General Regression Neural Networks (GRNN) are a class of RBNN used predominantly for non-linear regression [19]. The hidden layer is identical in structure and setup to the standard RBNN with a Gaussian kernel centered around each exemplar in the training set. There is an additional layer, as well as an additional output from the hidden layer. The eventual output of this network is the weighted output (z) scaled by the unweighted output of the hidden layer (s). Figure 2-7 illustrates this architecture.

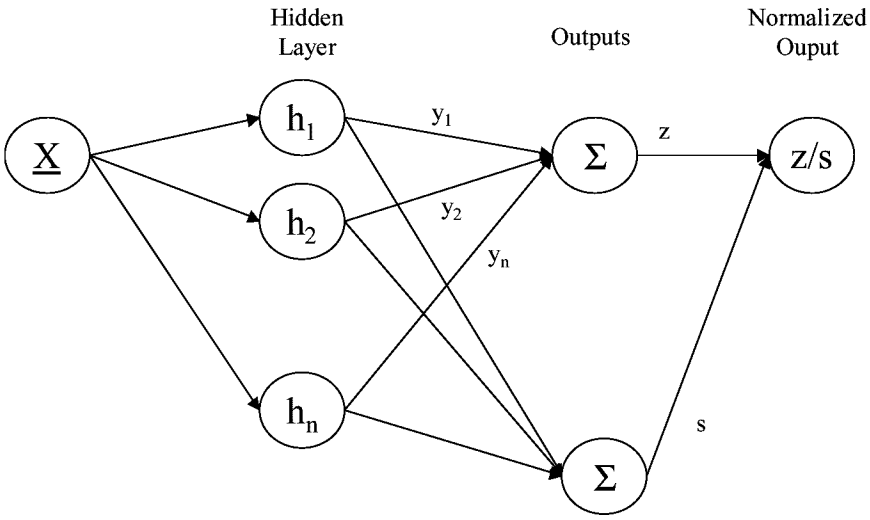

Figure 2-7. GRNN with Single Output

The primary difference between GRNN and RBNN is the training of the hidden weights. There is no training for GRNN's [19]. Each ($\underline{x}_i, y_i$) pair in the training set is folded into the network. The input vector, $\underline{x}_i$, is the center of radial basis function, h_i, and the output, y_i, is the hidden weight for that node. If the spread, σ_i, is very small, the network will have no error against the training set, however, the network will not be applicable to new exemplars. The choices for σ_i can be made in the same manner as the RBNN, using the nearest neighbor method.

2.5 Evaluation Techniques

There are several common techniques used to evaluate the utility of a classifier. The most common is estimating the Actual Error Rate (AER). This estimate of true error is obtained by applying the classifier to an independent validation set. This is due to the fact that using the training set will tend to underestimate the error [3]. There are two components to error, namely False Positive (FP) and False Negative (FN). Positive corresponds to the target, Class 1 and negative relates to the clutter, Class 2. A Confusion Matrix displays this information graphically as depicted in Figure 2-8.

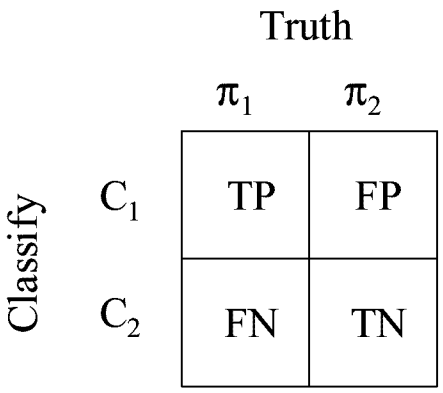

Figure 2-8. Confusion Matrix [3]

AER can be computed directly from a CM

$$AER = \frac{FP + FN}{TP + FP + FN + TN} \tag{2.34}$$

Using the estimate of AER to compare two classifiers can produce misleading results, particularly if the prior probabilities are very different [1]. Figure 2-9 illustrates two different classifiers applied to a notional data set. Classifier 1 has the smaller AER (95% vs. 94%), and would be considered the best classifier based on this measure. However, everything is classified as Class 2, and nothing is detected. No classifier is required to produce this output, an individual can simply assign Class 2 membership to every

exemplar. Classifier 2 has only a much better probability of detection (80% vs 0%), defined to be

$$P_D = \frac{TP}{TP + FN} \tag{2.35}$$

where *TP* and *FN* are defined as in Figure 2-7. Classifier 2 also has an only slightly higher probability of false alarm (5% vs. 0%), defined as

$$P_{FA} = \frac{FP}{TN + FP} \tag{2.36}$$

With this information, Classifier 2 appears to be the better classifier.

Classifier 1

	π_1	π_2
C_1	0	0
C_2	5	95

Classifier 2

	π_1	π_2
C_1	4	5
C_2	1	90

Figure 2-9. CM Comparison for Notional Data

2.5.1 Receiver Operating Characteristic Curves

The CM (as well as AER) only address the performance of the classifiers at the optimal decision threshold. Receiver Operating Characteristic (ROC) curves plot P_{FA} against P_D for different decision thresholds [1]. Figure 2-10 illustrates the general construction of the curve. The decision threshold is set at a given number of intervals across the range of the classifiers output. As the threshold changes from left to right (in this figure), both the P_{FA} and P_D increase as fewer exemplars are classified as Class 1.

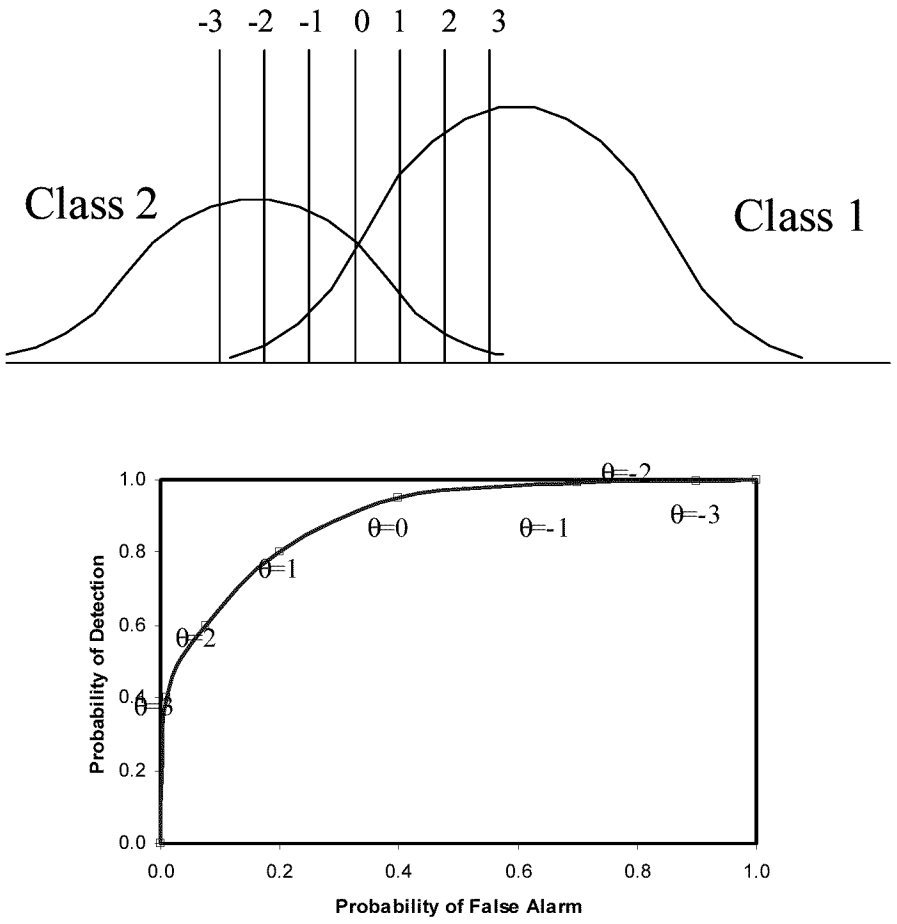

Figure 2-10. ROC Curve and Decision Thresholds

There are several metrics that can be used to evaluate ROC curves [1]. The first is by visual inspection. If two (or more) ROC curves are overlaid and one curve is always higher (a larger P_D for all P_{FA}), this classifier performs better. This will work in distinguishing classifiers, provided there is no overlap. In the latter, more common circumstances, objective metrics are necessary.

Alsing [1] presents a metric that can be used to objectively compare overlapping ROC curves, namely mean distance metric. ROC curves are compared to the chance line, which passes from the origin to (1,1). This line represents the ROC curve for random

classification. On this curve, the $P_D = P_{FA}$ for all decision thresholds, and corresponds to the value θ used to generate the point on the ROC Curve. The metric is the average distance of the ROC curve against this line for all points used to generate the ROC curve. In practice, this metric is

$$MD = \frac{\sum_{i=1}^{n} \left\| (P_D(\theta_i), P_{FA}(\theta_i)) - (\theta_i, \theta_i) \right\|_1}{n} \tag{2.37}$$

where $P_D(\theta_t)$ and $P_{FA}(\theta_t)$ are the ordered pair of the ROC curve based on the i^{th} decision threshold θ_t. The classifier with the largest mean distance metric is considered to perform best for the specific problem.

2.5.2 Multinomial Selection Procedures

Alsing [1] developed another comparison procedure, a Multinomial Selection Technique. This technique compares posterior probabilities for each point in the validation set. The posterior probabilities for quadratic discriminant analysis applied to a two class problem are [3]

$$PP_i = \frac{d_{Q_i}}{d_{Q_1} + d_{Q_2}}. \tag{2.38}$$

For FFNN that are trained to zero and one, the class one posterior probability for a given exemplar is simply the network output. The class two posterior probabilities for the same exemplar are one minus the output [3]. The posterior probabilities for a RBNN are more problematic. Unlike FFNN using a sigmoid in the output layer, the outputs for RBNN are not restricted to the interval (0,1). The outputs therefore are normalized to the interval [0,1], and these normalized outputs become the posterior probabilities.

Once the posterior probabilities have been calculated, the multinomial statistic can be calculated. For each exemplar in the validation set, a "win" is given to the

classifier with the highest posterior probability for the class to which the exemplar belongs. When the entire validation set has been processed, the multinomial statistic for each classifier becomes the number of "wins" divided by the total number of validation points. These statistics are estimates of the true multinomial probabilities, and confidence intervals can be created around each value. If the confidence intervals for two different classifiers do not overlap, the classifier with the larger multinomial statistic can be determined to be a better classifier for the problem. According to Alsing [1], this can be used if the other metrics described above fail to determine the best classifier.

In this chapter, three different classifiers were explored: DA, FFNN, and RBNN. Feature selection techniques were described for DA and FFNN. Additionally, means to evaluate the performance of these classifiers were discussed: AER, ROC metrics and the multinomial selection procedure. In the next chapter, a feature selection technique will be developed for RBNN in addition to a new clustering routine.

3 Radial Basis Neural Network Techniques

3.1 Overview

This chapter introduces two new techniques, derivative based saliency (DBS) and signal-to-noise ratio (SNR^{RBNN}) clustering. The first section of this chapter details DBS as a feature selection technique for Radial Basis Neural Networks (RBNN's). DBS will be compared in Experiment 3-1 with the feature selection techniques used with Discriminant Analysis (DA) and Feed Forward Neural Networks (FFNN), discriminant loadings and Signal-to-Noise Ratio (SNR) respectively. The second section describes the SNR^{RBNN} clustering algorithm. SNR^{RBNN} will be compared with K-Means and the Radial Basis Function Iterative Construction Algorithm (RICA) in Experiment 3-2. The final section develops the iterative architecture and feature selection algorithm. This algorithm will be compared to discriminant loadings and SNR in Experiment 3-3, a repeat of Experiment 3-1 with the integrated algorithm replacing *K*-Means.

3.2 Derivative Based Saliency

A derivative based saliency measure appears to be the only feature selection available for RBNN's. Weight-based saliency measures are inappropriate because the weights are not applied directly to the features as in FFNN. As with FFNN, it is necessary that the data be standardized so that a unit change in each feature is equivalent. Otherwise, it is likely the feature with the highest variance will have the highest measure. The network output for a given exemplar i is

$$z^{(i)} = \sum_{j=1}^{p} w_j \exp\left[\frac{-1}{2\sigma_j^{\,2}} \sum_{k=1}^{m} \left(x_k^{\,(i)} - \mu_k^{\,(j)}\right)^2\right] \tag{3.1}$$

where p is the number of centers, m is the number of features and $\mu_k^{(j)}$ is the k^{th} component of the j^{th} center. The partial derivative of the network output of exemplar i with respect to feature k is

$$DS_{ik} = \frac{\partial z^{(i)}}{\partial x_k} = \sum_{j=1}^{p} \frac{-w_j}{\sigma_j^2} \left(x_k^{(i)} - \mu_k^{(j)} \right) h_{ij} \tag{3.2}$$

where

$$h_{ij} = \exp\left[\frac{-1}{2\sigma_j^2} \left(\underline{x}^{(i)} - \underline{\mu}^{(j)} \right)^T \left(\underline{x}^{(i)} - \underline{\mu}^{(j)} \right) \right]. \tag{3.3}$$

When taking the mean saliency across all the exemplars, the average of DS_{ik} can be misleading. Different exemplars, particularly in different classes, can have opposite signs, moving the measure two zero. The measure of interest is the magnitude of the measure across the exemplars. Therefore, the mean absolute saliency measure for the k^{th} feature is

$$MS_k = \frac{1}{n} \sum_{i=1}^{n} \left| DS_{ik} \right| \tag{3.4}$$

where n is the number of exemplars in the training set. Figure 3-1 illustrates the algorithm in flow-chart format. The complete derivation is provided in the Appendix.

Examination of Equations (3.2) and (3.3) seem to indicate that prior clustering of centers will improve the performance of the measure. If no clustering is performed, the n exemplars act as centers. Equation (3.2) will evaluate to zero (or approach it) for most of the exemplar center pairs. For $i = j$, $x_k^{(i)} - \mu_k^{(j)} = 0$, and for those exemplars far from centers, h_{ij} will approach zero. If exemplars are represented by a center close to them, such as the mean, neither part of the equation will approach zero.

3.2.1 Experiment 3-1: Simple Feature Selection Test

This supposed difference must be verified, and this technique for feature selection needs to be compared against discriminant loadings and SNR. A simple problem will be used to provide preliminary answers, and also explore the effect noise has on classification problems. The training and validation sets for this problem are randomly generated according to the following distributions. Feature 1 is normally distributed with a standard deviation of one, and a mean of one for class one and a mean of negative one for class two. This is the only true feature in the problem, but there is considerable overlap between the two populations. The remaining nine features are noise features, with all data distributed uniformly between negative one and one. Each training set consists of eleven exemplars, and each validation set of fifty exemplars from each class. Feature selection is performed against the training set, and the error rate is computed on the validation set. Four classifiers (and feature selection techniques) were evaluated against this problem: DA with discriminant loadings, FFNN with SNR, RBNN with no clustering and DBS and RBNN with *K*-means clustering and DBS. Fifty random samples of both training and validation sets were made, and the average performance is reported.

Figure 3-2 illustrates the relationship between classification accuracy and the number of noise features. The first conclusion that can be made is noise adversely impacts classification accuracy for all the competing classifiers, and this difference is statistically significant for an overall $\alpha = 0.1$. This is most true of DA, which performs considerably worse than the artificial neural networks with all the noise variables included, but which performs best with only one feature remaining. Table 3-1 and Figure 3-3 explain a large part of why this is true. DA and Discriminant Loadings did not make a single mistake in retaining Feature 1 until the end. Table 3-1 includes confidence interval half-widths with an overall $\alpha = 0.1$ using the Bonferroni approach. Clustering improves the performance of DBS applied to the RBNN's, validating the premise of the

feature selection technique. However, even with *K*-Means clustering, DBS falls well short of the performance of FFNN with SNR. This leads to poor classification accuracy when more features are removed. This can be seen in Figure 3-2. The classification accuracies for both FFNN and the RBNN with *K*-Means clustering are approximately equal with four features remaining. After this point, the FFNN continues to improve, while the RBNN begins to plateau, and then dramatically worsens for one feature remaining. This gradually worsening performance is caused by the RBNN removing the good feature too early and too often.

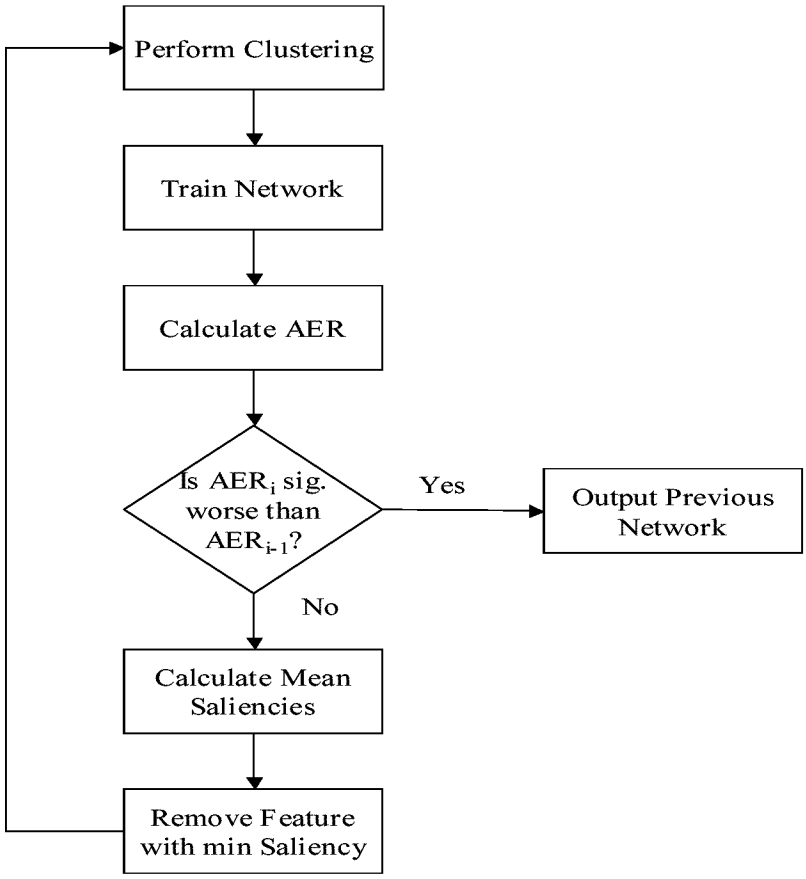

Figure 3-1. DBS Iterative Feature Selection Algorithm

Table 3-1. Results of Feature Selection Test

Measures	DA	FFNN	RBNN w/o clust	RBNN w/ K-Means
Average Ranking, Feature 1	1	1.04	1.74	1.4
Proportion Feature 1. Ranked First	1	0.96	0.44	0.76
90% CI Half-Width	0.0582	0.0621	0.1573	0.1354

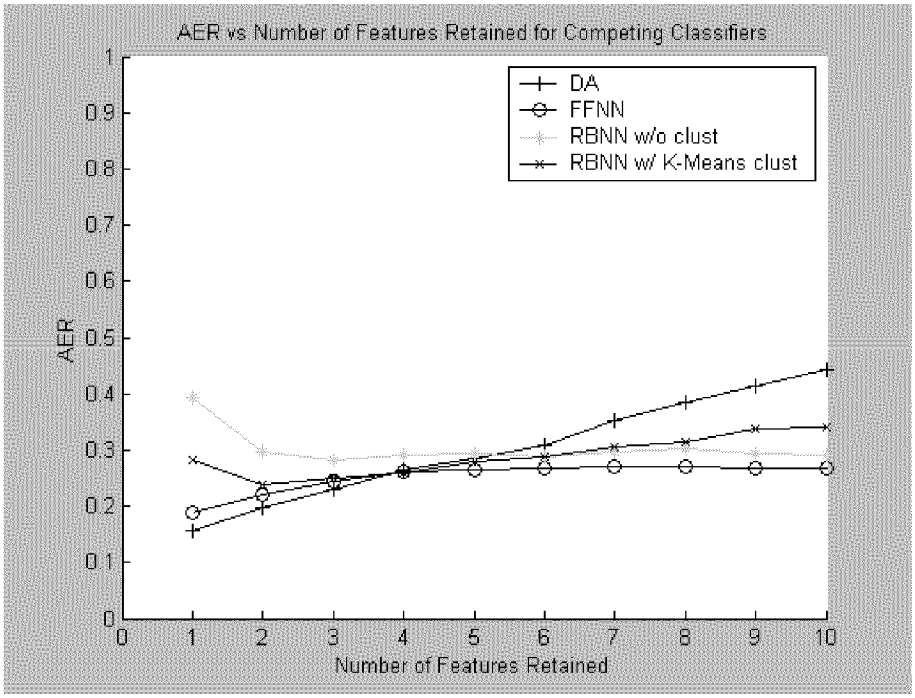

Figure 3-2. AER for Experiment 3-1

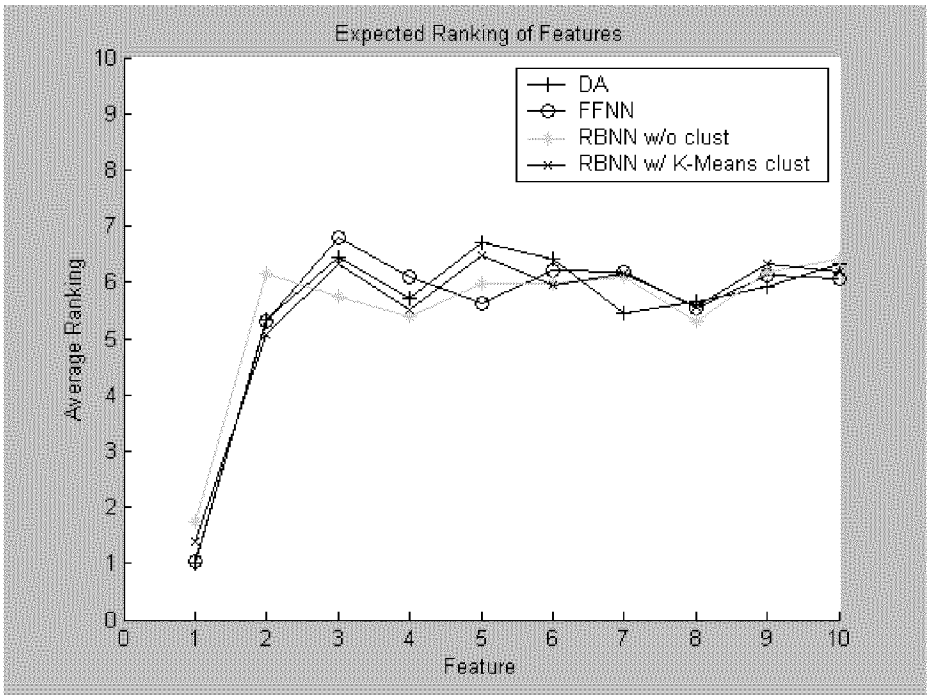

Figure 3-3. Average Feature Rankings Experiment 3-1

3.3 SNR Clustering Technique

The next topic of discussion involves using the RBNN itself to perform clustering for RBNN. This SNR approach follows the same basic approach used in SNR for feature selection in FFNN. The first requirement is a noise variable. For feature selection this involves a noise feature. In clustering, this will require a noise center added to the RBNN. Before defining what a noise center is, the signal-to-noise ratio measure will be defined. As with the SNR used for feature selection, the weights of features will be compared to the weights of the noise variable. In the clustering instance, the noise is defined as

$$Noise = \left(w_{p+1}\right)^2 \tag{3.5}$$

where p is the number of centers in the original problem. The SNR measure for each center under consideration is

$$SNR_j^{RBNN} = 10\log_{10}\left(\frac{w_j^2}{Noise}\right). \tag{3.6}$$

The superscript RBNN is used to distinguish this from the SNR used for feature selection in FFNN. Any center with a signal-to-noise ratio less than zero is considered to be noise and unnecessary.

The SNR measure is very straightforward, but what is not obvious is the meaning of noise as it applies to a center. When data are standardized to mean zero and unit variance, most of the data will be massed in the region between one and negative one in each feature. In this thesis, the noise center will be defined as a random vector from this region. The center will be distributed uniformly between negative one and one for each feature. If a random center made with no knowledge of the problem has a greater impact on the output (i.e., has a larger weight) than other centers, they can be considered as noise.

The SNR clustering algorithm proceed as follows. The RBNN is first trained using each exemplar as a center with a noise center added. When the training is complete, the SNR measures are calculated for each center. Those centers with negative ratios are clustered with the nearest within-class center with a positive SNR. The centers for the final network become the cluster means and the network is trained using these centers. Figure 3-4 further illustrates the algorithm.

3.3.1 Experiment 3-2: Block-C Clustering Test

This clustering technique will be compared with *K*-Means and RICA in the following example. The data sets will be generated from the Block-C distribution shown in Figure 1-1. Each training set will contain 60 randomly generated data, while each validation set will be made of 100. All three clustering algorithms will be applied to the training data. Receiver Operating Characteristic (ROC) curves and estimates of the AER

will be generated from the validation set. Thirty replications of this procedure will be performed (with a different random center generated for each iteration), and the averages across the replications reported.

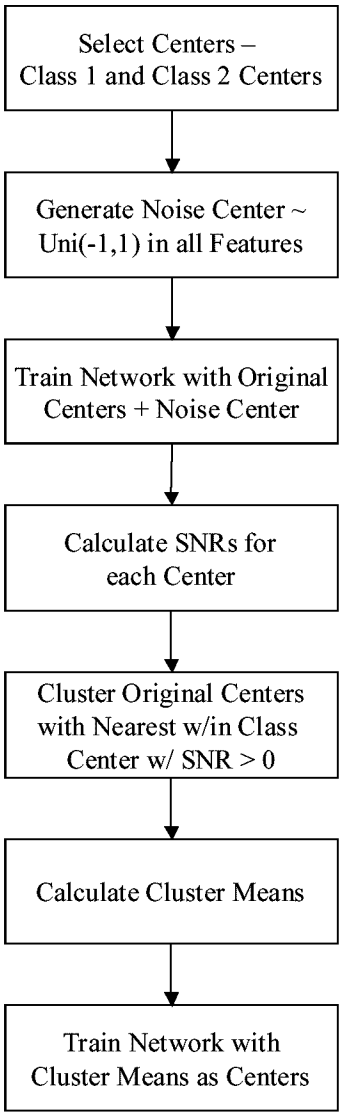

Figure 3-4. SNR^{RBNN} Clustering Algorithm

Figure 3-5 displays the average ROC curves for the three clustering algorithms. *K*-Means clearly dominates the other two clustering techniques for this problem. The same experiment was run with 120 data points in each training set to examine the

performances with more data for training. Figure 3-6 demonstrates that SNR^{RBNN} performs almost identically to *K*-Means. The AER for K-Means is slightly better than for SNR^{RBNN} (0.1117 compared to 0.1173), but is not statistically significant. RICA improves but is still dominated by the other two techniques.

While SNR^{RBNN} performs as well as *K*-Means with 120 data points in the training set, this problem illustrates the shortcomings of this clustering technique as it was applied to this problem. To perform the clustering, training was accomplished first with all the exemplars as centers and then an additional network was trained with the reduced centers. This can quickly increase the number of calculations required, particularly as the sample sizes increase. If the network is trained with no clustering, why cluster and train the network again? The next section will discuss how SNR^{RBNN} can be applied in an iterative manner.

3.4 An Integrated Architecture and Feature Selection Algorithm

As discussed in Section 3.3, applying SNR^{RBNN} to a problem where clustering will be done only once entails redundant labor. While it will produce a more parsimonious model, *K*-Means will accomplish this with less computational effort. If however, clustering must be done repeatedly to support feature selection, it might prove useful. One of the reasons *K*-Means performs erratically with DBS is that different centers are generated for each iteration. This section will propose an iterative feature selection algorithm, and test it against the same problem analyzed in Experiment 3-1. Steppe *et. al.* [16] provide the basis for an alternating architecture and feature selection approach for FFNN. The removal of a hidden node was performed followed by a removal of a feature. This process was repeated until the appropriate number of hidden nodes and features were selected.

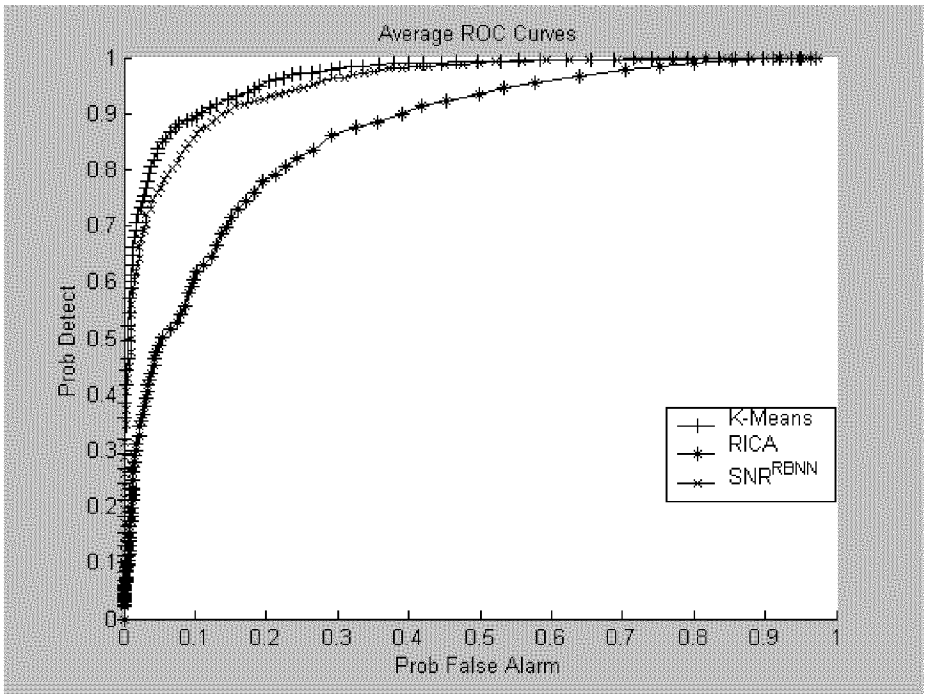

Figure 3-5. Block C Clustering Test – 60 Data Points

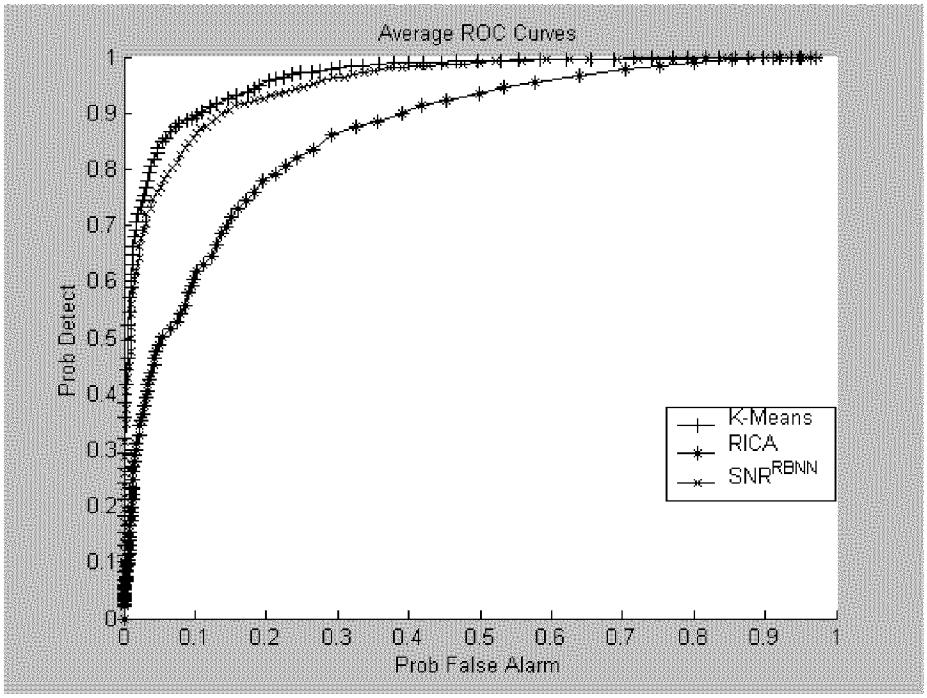

Figure 3-6. Block C Clustering Test – 120 Data Points

This algorithm follows the basic approach of DBS. The first iteration begins with SNR^{RBNN} clustering performed with the whole training set starting as centers. Feature selection is performed, and the least significant feature is removed. The second, and each successive, iteration begins with the centers provided by the previous iteration, clustering the original centers with the nearest within-class retained center. SNR^{RBNN} is applied to the current set of centers (minus the removed feature). For each iteration, the computational effort is less, as each step entails training with fewer centers. Figure 3-7 describes the algorithm in more detail.

This algorithm can be very flexible, with *K*-Means being used to cluster for the first iteration if the training set is very large. While it is flexible, it does require supervision. If the classification accuracy drops significantly after an iteration, it could either indicate a true feature deletion or that necessary centers have been removed. At this point, the centers from the previous iteration could be retained, and feature selection can proceed without clustering until it is determined that only significant features remain.

3.4.1 Experiment 3-3: Simple Feature Selection Test Revisited

Figures 3-5 and 3-6 demonstrate the effectiveness of this clustering algorithm applied to the problem described in Section 3.2. The performance of SNR^{RBNN} used iteratively with feature selection performs as well as SNR applied to the FFNN and Discriminant Loadings used in DA. Table 3-2 illustrates this. The average feature rankings are identical, and SNR^{RBNN} made only one more mistake in ranking than SNR.

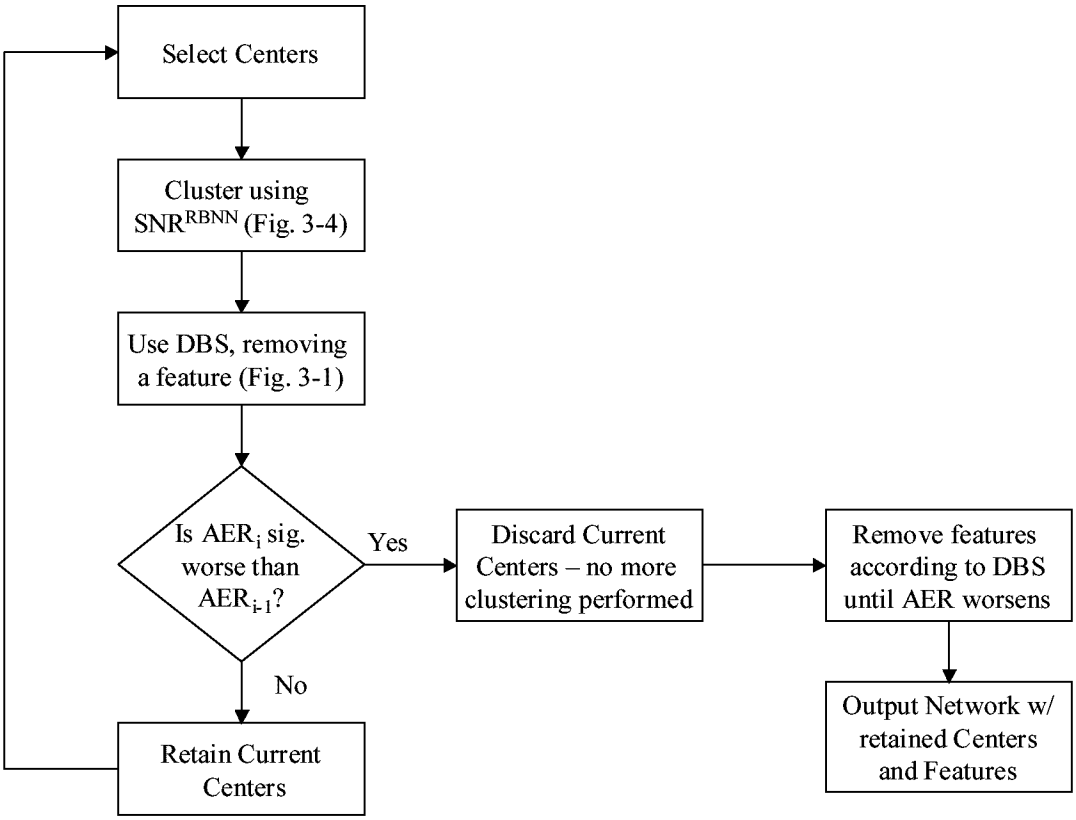

Figure 3-7. Integrated SNR^{RBNN}/DBS Feature Selection Algorithm

Table 3-2. Results of Feature Selection Test w/ SNR^{RBNN}

Measures	DA	FFNN	RBNN w/o clust	RBNN w/ SNR^{RBNN}
Average Ranking, Feature 1	1	1.06	1.72	1.06
Proportion Feature 1. Ranked First	1	0.96	0.62	0.94
95% CI Half-Width	0.0582	0.0621	0.1539	0.0753

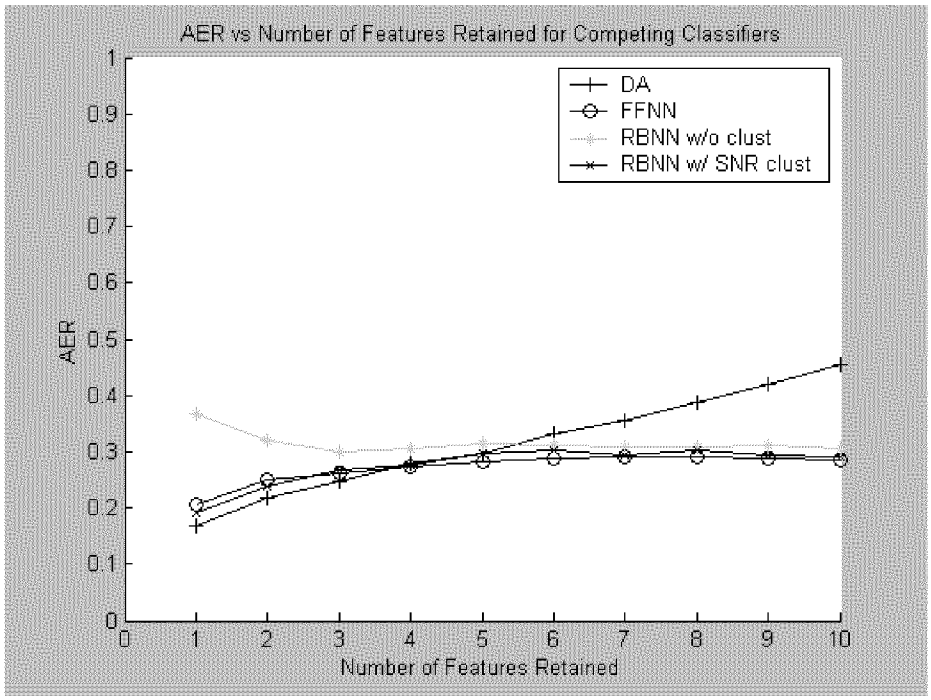

Figure 3-8. AER for DA, FFNN and RBNN with w/ SNR^{RBNN} and no clustering

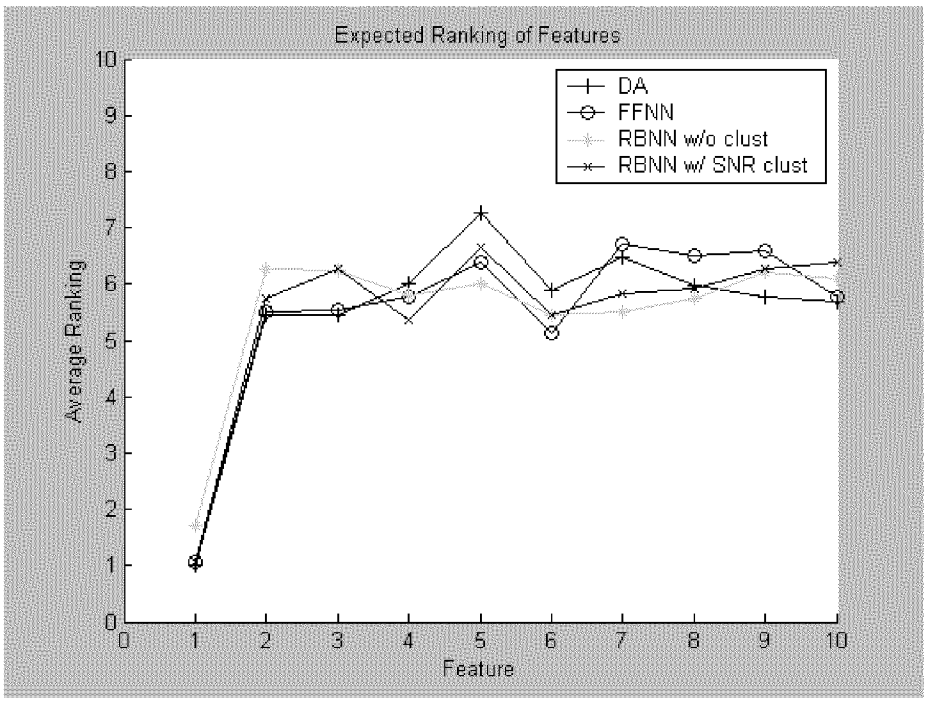

Figure 3-9. Average Feature Rankings for the Four Classifiers

This chapter has introduced two new techniques: derivative based saliency feature selection, and signal-to-noise ratio clustering. Without clustering, the feature selection routine does not perform well, even on the simple problem explored in Section 3.1. While the clustering algorithm performs fairly well, approaching the performance of *K*-Means as the sample size increases, it does not perform better. Also, for a single iteration, it requires redundant work (classification is performed twice). However, when the two techniques are coupled, they provide performance equivalent to Discriminant Loadings and SNR. These results are only for a simple problem, and more challenging problems will be addressed in the following chapter.

4 Evaluation of Competing Classifiers

4.1 Overview

This chapter will evaluate Discriminant Analysis (DA), Feed Forward Neural Networks (FFNN) and Radial Basis Neural Networks (RBNN) applied to several challenging problems. The first problem will be Block-C addressed in Sections 1.1 and 3.3. The second application will be the University of Wisconsin Breast Cancer data. The final application will be the classic Fisher's Iris Problem with noise features added. The purpose for these final two experiments is to evaluate the efficacy of the feature selection algorithms in addition to classifier performance. The analysis techniques in Section 2.5 will be used to compare the different classifiers.

4.2 Experiment 4-1: Block-C Classifier Test

DA, FFNN and RBNN will be applied to the Block-C problem. For the first experiment, 240 training points and 100 validation points will be used. Thirty iterations will be performed, with the average Receiver Operating Characteristic (ROC) curves, Apparent Error Rate (AER), multinomial test statistics and mean distance metrics being generated for each classifier. Figure 4-1 displays the average ROC curves, and Table 4-1 shows the average metrics for each classifier. RBNN will apply SNR^{RBNN} to perform clustering on the centers. The FFNN will use eight hidden nodes, and will use 40% of the training data for internal validation.

The RBNN with SNR^{RBNN} clustering significantly outperforms the other two classifiers in classification accuracy. Both Artificial Neural Networks (ANN) perform much better than DA (which performs worse than just guessing). This experiment was repeated for training set sizes of 480 and 960. DA and FFNN were applied identically,

while RBNN used *K*-Means with *k=100* for each class to cluster the centers for both experiments.

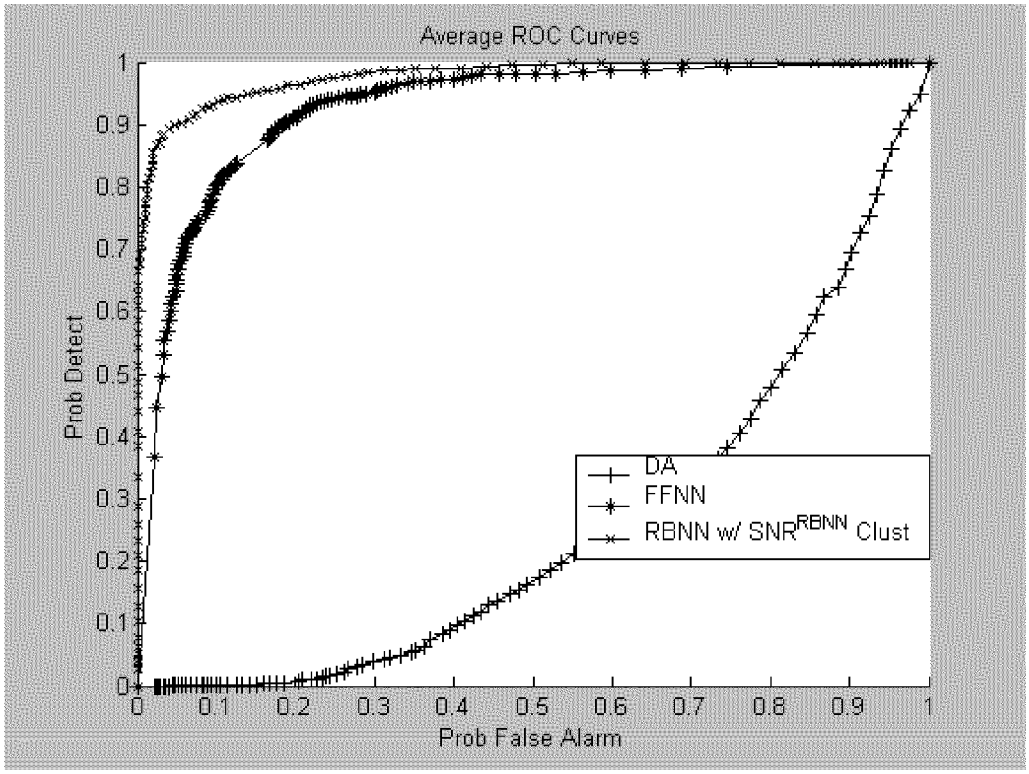

Figure 4-1. Average ROC Curves for Block-C Problem, 240 Training Points

Table 4-1. Average Metrics for Block-C Problem, 240 Training Points

Measures	DA	FFNN	RBNN
AER	0.643	0.1613	0.086
90% CI Half-Width	0.0461	0.0622	0.0125
Mean Distance	0.3437	0.6286	0.5385
90% CI Half-Width	0.0383	0.084	0.0156
Multinomial	0.1377	0.615	0.2473
90% CI Half-Width	0.0193	0.1028	0.0948

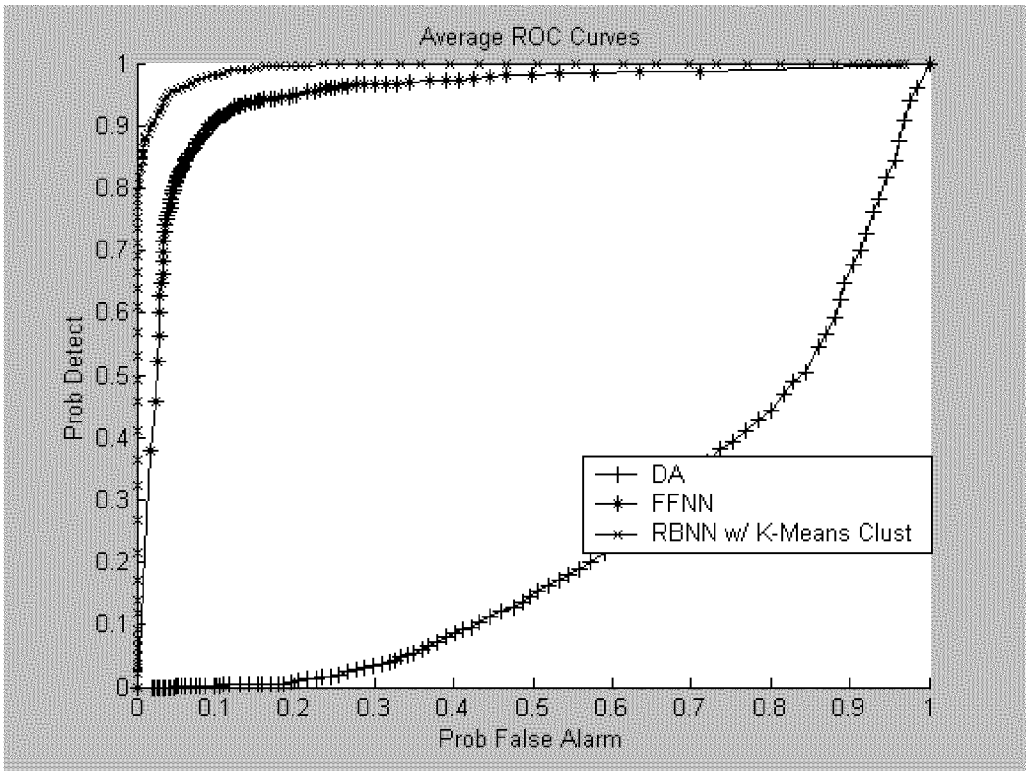

Figure 4-2. Average ROC Curves for Block-C Problem, 480 Training Points

Table 4-2. Average Metrics for Block-C Problem, 480 Training Points

Measures	DA	FFNN	RBNN
AER	0.689	0.0967	0.046
90% CI Half-Width	0.0263	0.0152	0.0095
Mean Distance	0.3629	0.7166	0.6126
90% CI Half-Width	0.0273	0.0348	0.0168
Multinomial	0.112	0.6863	0.2017
90% CI Half-Width	0.0105	0.0422	0.0181

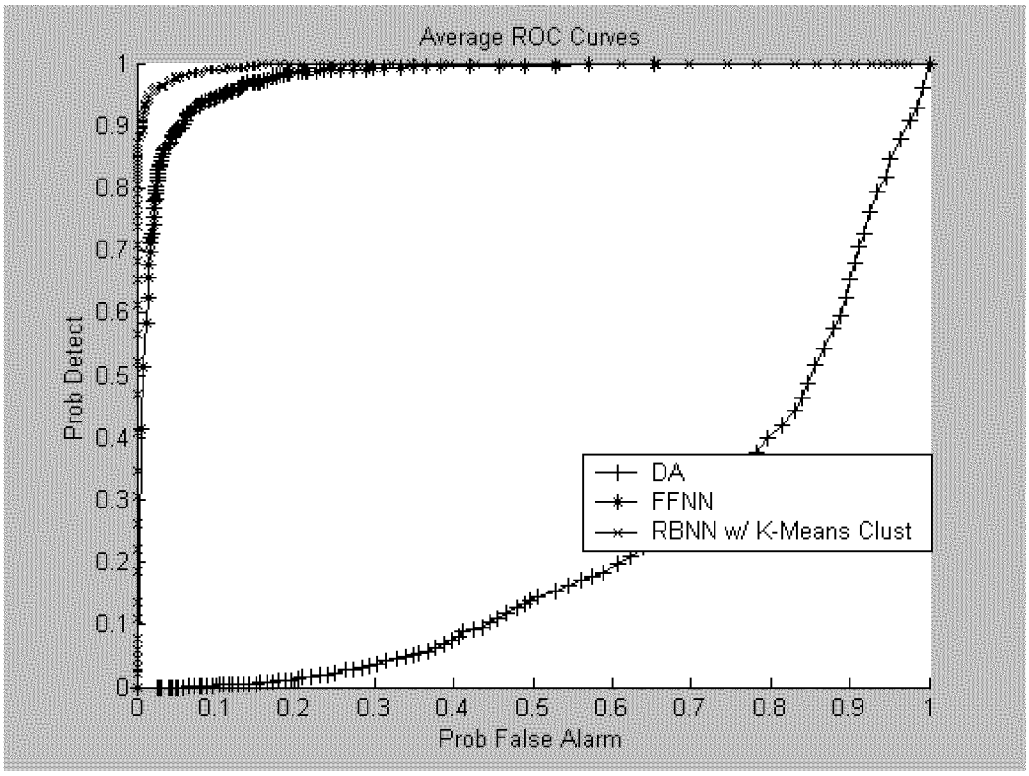

Figure 4-3. Average ROC Curves for Block-C Problem, 960 Training Points

Table 4-3. Average Metrics for Block-C Problem, 960 Training Points

Measures	DA	FFNN	RBNN
AER	0.706	0.076	0.027
90% CI Half-Width	0.0303	0.0179	0.0083
Mean Distance	0.348	0.7435	0.6247
90% CI Half-Width	0.0143	0.0424	0.0205
Multinomial	0.0993	0.686	0.2147
90% CI Half-Width	0.0119	0.0472	0.0493

Figures 4-2 and 4-3 display the respective ROC curves, and Tables 4-2 and 4-3 show the metric performance. The domination in ROC curves and AER continue for the RBNN, although the FFNN appears to be converging. The other metrics however, identify the FFNN as the better classifier. For all three sample sizes, the FFNN has a higher mean distance metric, although for 240 training points the difference is not significant with an overall $\alpha = 0.1$. The FFNN also perform significantly better in the multinomial selection metric for all sample sizes.

4.2.1 Experiment 4-2: Perturbed Block-C Classifier Test

Alsing [1] asserts that a classifier that performs better for mean metric distance will be more robust to perturbations in the data. Under this hypothesis, the FFNN will better handle changes in the data than the RBNN. To test this, the three experiments conducted in Section 3.2 were repeated with the validation data perturbed. The validation data were shifted 0.1 in both dimensions. Figures 4-4, 4-5 and 4-6 show the averages ROC curves for the three classifiers applied to the different sample sizes. Tables 4-4, 4-5 and 4-6 show the average metrics for the three experiments.

For the training size of 240 exemplars, the mean distance metric was not significantly different for RBNN and FFNN. While not statistically significant, FFNN still performed better in this metric. Figure 4-4 and Table 4-4 show that the FFNN reacted better to the perturbed data. The difference in AER is no longer significant, and the ROC curves now overlap. Although the mean distance is still not significant, the multinomial statistic is significant. It is concluded that the FFNN is the best classifier for this perturbed problem.

This performance is repeated for the sample sizes of 480 and 960. Figures 4-5 and 4-6 show that the ROC curves for the FFNN now dominate the RBNN curves. Tables 4-5 and 4-6 show the AER is no less for the FFNN, although it is still statistically

insignificant. Both the mean distance metric and the multinomial statistic indicate that FFNN performs better than the RBNN. It is concluded that the FFNN is more robust to perturbations in the validation data, and is a better classifier for the perturbed problem.

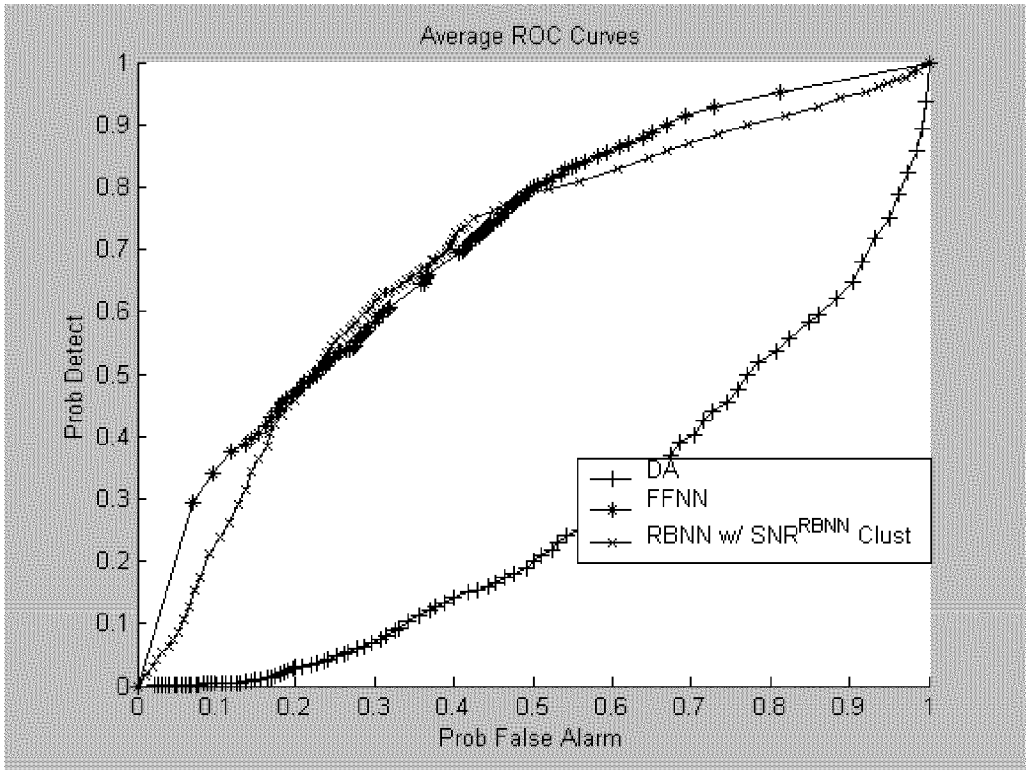

Figure 4-4. Average ROC Curves for Perturbed Block-C, 240 Training Points

Table 4-4. Average Metrics for Perturbed Block-C, 240 Training Points

Measures	DA	FFNN	RBNN
AER	0.672	0.338	0.323
90% CI Half-Width	0.0268	0.0257	0.0243
Mean Distance	0.3501	0.3823	0.2284
90% CI Half-Width	0.0383	0.0506	0.0135
Multinomial	0.324	0.4877	0.1883
90% CI Half-Width	0.0226	0.0517	0.0556

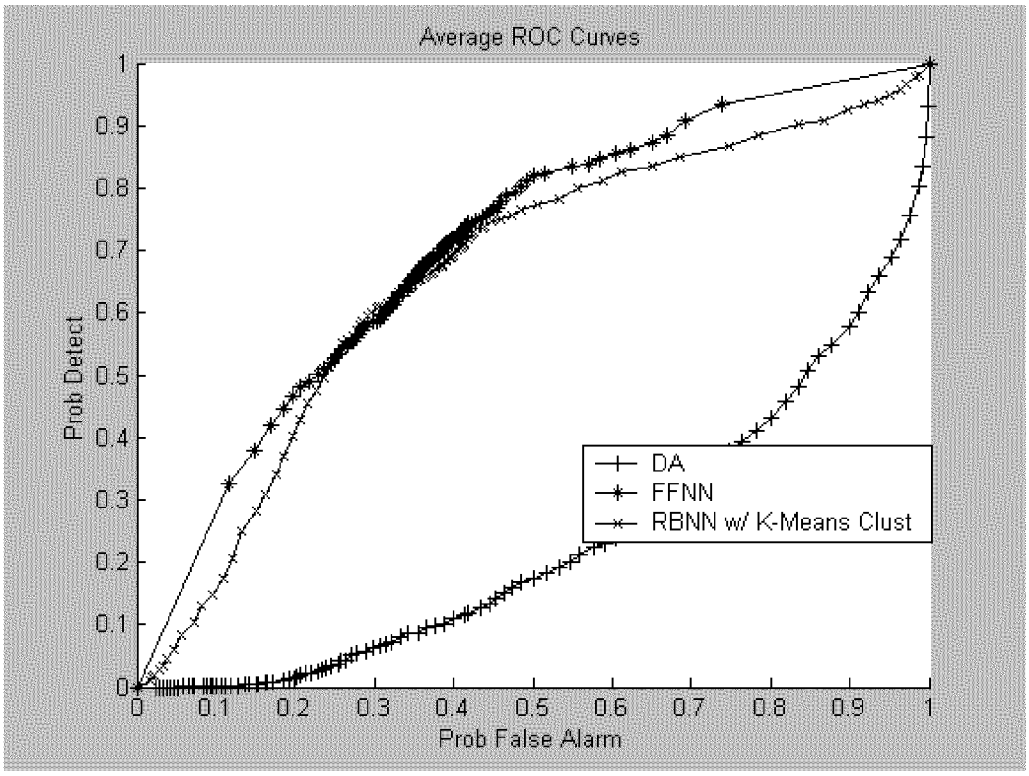

Figure 4-5. Average ROC Curves for Perturbed Block-C, 480 Training Points

Table 4-5. Average Metrics for Perturbed Block-C, 480 Training Points

Measures	DA	FFNN	RBNN
AER	0.6903	0.33	0.3367
90% CI Half-Width	0.0300	0.0187	0.0174
Mean Distance	0.428	0.4121	0.2823
90% CI Half-Width	0.0435	0.0343	0.0217
Multinomial	0.3603	0.5157	0.124
90% CI Half-Width	0.0157	0.0313	0.0309

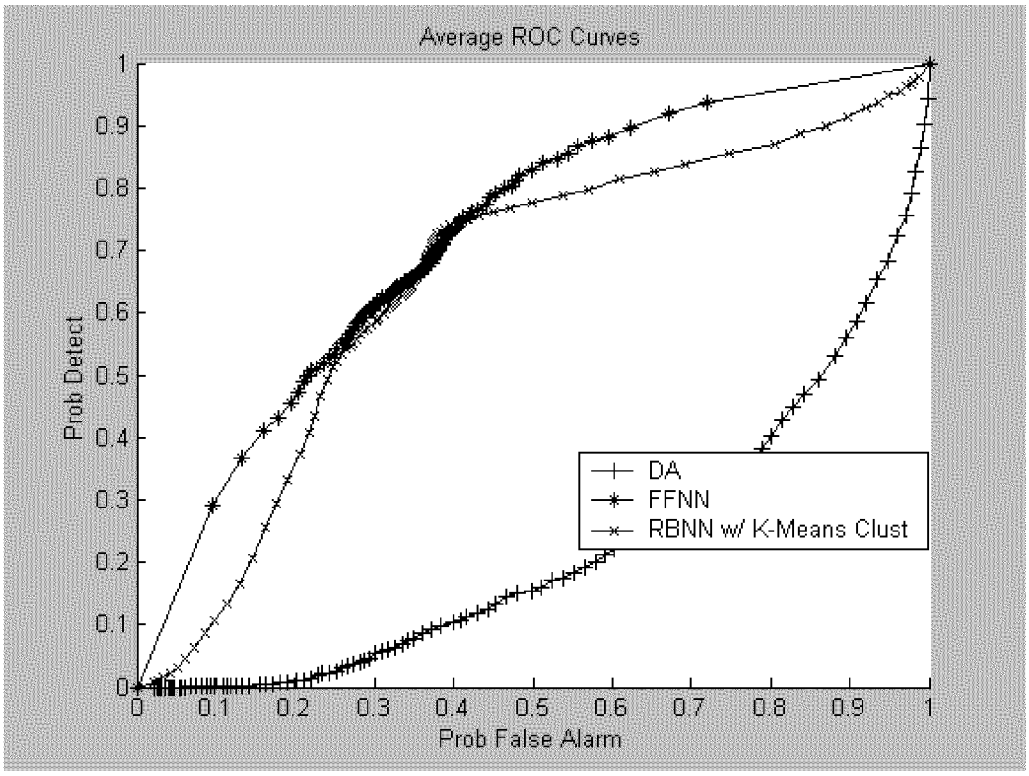

Figure 4-6. Average ROC Curves for Perturbed Block-C, 960 Training Points

Table 4-6. Average Metrics for Perturbed Block-C, 960 Training Points

Measures	DA	FFNN	RBNN
AER	0.6957	0.3443	0.353
90% CI Half-Width	0.0214	0.0205	0.0214
Mean Distance	0.4313	0.4252	0.2674
90% CI Half-Width	0.0215	0.029	0.0318
Multinomial	0.376	0.5027	0.1213
90% CI Half-Width	0.0236	0.0281	0.0220

4.3 University of Wisconsin Breast Cancer Data

The University of Wisconsin Breast Cancer Data (UWBCD) set obtained from the University of California-Irvine [18] consists of 699 tissue samples. 241 exemplars were malignant (Class 1) and 458 were benign (Class 2). Each exemplar contained nine features: clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nuclei, and mitoses. Alsing [1] produced feature rankings by applying SNR to the data. Bare nuclei and cell thickness were the most significant, and mitoses and single epithelial cell size were the least significant.

4.3.1 Experiment 4-3: UWBCD Classifier Comparison

For this experiment, the three classification techniques were applied to the data set to include all nine features. The training set consisted of 350 exemplars, with 349 exemplars held out for the validation set. The FFNN used 18 hidden nodes and partitioned the training set into 210 training and 140 training test exemplars. The RBNN used SNR^{RBNN} to cluster the data which reduced the number of centers from 350 to 240.

Figure 4-7 displays the ROC Curves for the three classifiers and Table 4-7 shows the metrics for this experiment. Analysis of the ROC Curve and the AER yields no significant difference between the classifiers. There is no significant difference between the FFNN and DA for the multinomial selection metric, but both perform significantly better than the RBNN. The FFNN does perform significantly better than both the RBNN and DA for the mean distance metric and should be more robust to perturbations.

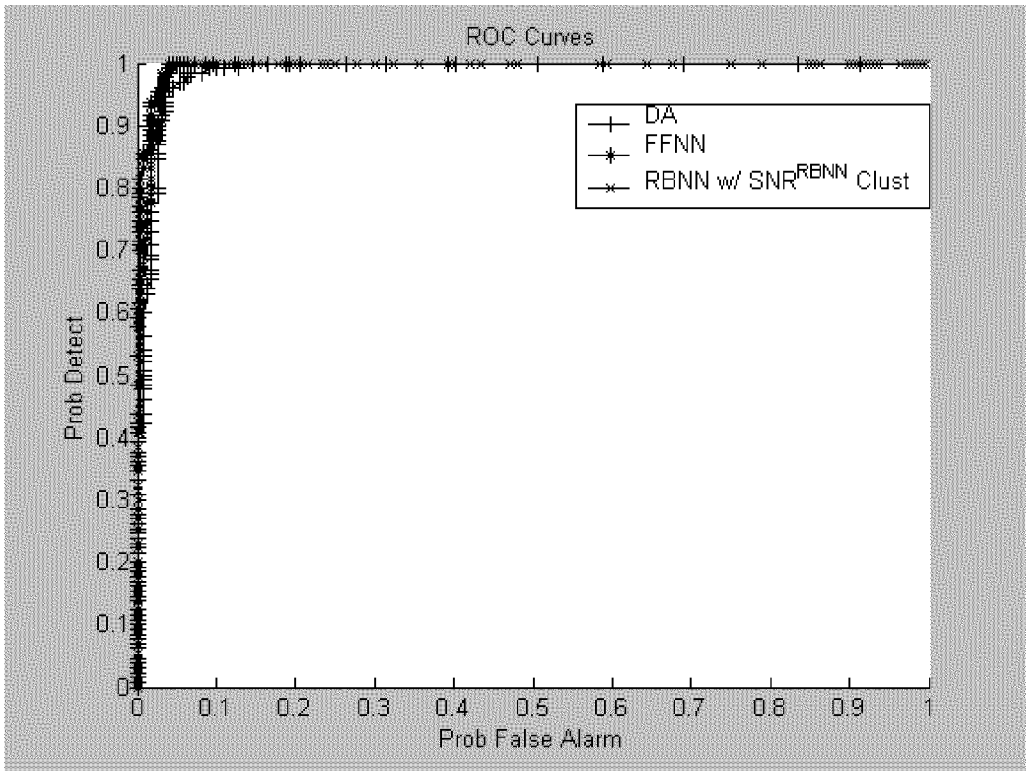

Figure 4-7. ROC Curves, UWBCD, 9 Features

Table 4-7. Metrics, UWBCD, 9 Features

Measures	DA	FFNN	RBNN
AER	0.0372	0.043	0.0372
90% CI Half-Width	0.0243	0.0260	0.0243
Mean Distance	0.6334	0.9129	0.6659
90% CI Half-Width	0.0361	0.0164	0.043
Multinomial	0.5043	0.4585	0.0372
90% CI Half-Width	0.0641	0.0639	0.0243

4.3.2 Experiment 4-4: Perturbed UWBCD Classifier Comparison

This next experiment tests the hypothesis that the FFNN will be more robust by perturbing the validation data set. The perturbation was accomplished by adding random draws from a normal population with mean zero and standard deviation of two to bare nuclei and clump thickness for each exemplar in the validation set. The partitioning of the data and the application of the classifiers was identical to Experiment 4-3. Figure 4-8 illustrates the ROC Curves and Table 4-8 displays the metric performance for the three classifiers against this perturbed data. The FFNN clearly dominates the RBNN and DA in all categories. The FFNN was decidedly more robust to the changes in the validation set.

4.3.3 Experiment 4-5: UWBCD Feature Selection Test

For this last experiment, seven features were added to the data set. Five features were noise variables uniformly distributed between zero and one. The remaining two additional features were redundant features, being slight modifications of two existing features, bare nuclei, a significant feature, and mitoses, a relatively insignificant feature. These features were slightly perturbed to allow for DA to work. If the features were identical, the inverse of the covariance matrix would not exist, and DA could not be applied. These feature were modified by adding random draws from a Normal(0,0.04) population to each exemplar's features.

The three feature selection techniques, Discriminant Loadings, signal-to-noise ratio (SNR) and derivative-based saliency coupled with SNR^{RBNN} clustering were applied to the data. Classification was performed as each feature was removed. Figure 4-9 shows the AER plotted against the number of features remaining. The minimum AER was chosen as the ideal termination point for each classifier, and the resultant ROC curves and metric performance are given in Figure 4-10 and Table 4-9.

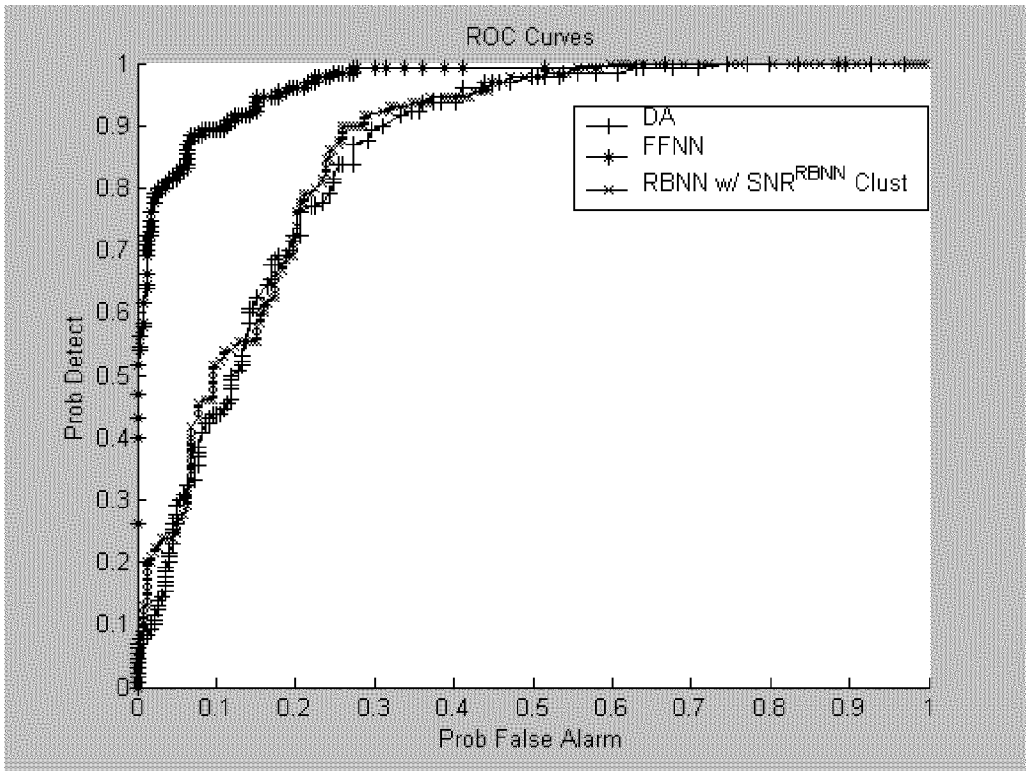

Figure 4-8. ROC Curves, Perturbed UWBCD, 9 Features

Table 4-8. Metrics, Perturbed UWBCD

Measures	DA	FFNN	RBNN
AER	0.3438	0.0917	0.2292
90% CI Half-Width	0.0609	0.0370	0.0539
Mean Distance	0.5504	0.7537	0.4384
90% CI Half-Width	0.0319	0.0141	0.0306
Multinomial	0.1318	0.8052	0.063
90% CI Half-Width	0.0433	0.0508	0.0311

DA terminated with four features remaining. All the noise features were removed, but six of the real features were also removed. The addition of the new features caused DA to perform significantly worse, even at its optimal point. The FFNN fared much better, removing all five noise features. Only one original feature was retained, mitoses, and its removal did not impact classification accuracy. The RBNN retained three noise features and both redundant features at its terminating point of ten features retained. At this point, significant features were removed prior to the removal of the noise features. SNR^{RBNN} clustering was performed for the first two iteration before further clustering affected classification accuracy. The number of centers was first reduced to 224 and finally to 75.

The FFNN and the RBNN were not significantly impacted by the noise and redundant features. The AER with all 16 features included is not significantly worse than at their optimal point for both networks. Both networks perform significantly better than DA at its optimal point. There are no significant differences between the ROC Curves and AER for the FFNN and the RBNN. However, the FFNN performs significantly better in the mean distance and multinomial selection metrics. The FFNN performs feature selection best, and is also the best classifier for Experiment 4-5.

Table 4-9. Metrics for UWBCD Feature Selection Test

Measures	DA	FFNN	RBNN
Features Retained	4	9	10
Noise Features Retained	0	0	3
Redundant Features Retained	1	1	2
AER	0.1289	0.0487	0.0458
90% CI Half-Width	0.0429	0.0276	0.0268
Mean Distance	0.5961	0.7976	0.6042
90% CI Half-Width	0.0342	0.0306	0.0421
Multinomial	0.0831	0.8539	0.063
90% CI Half-Width	0.0354	0.0453	0.0311

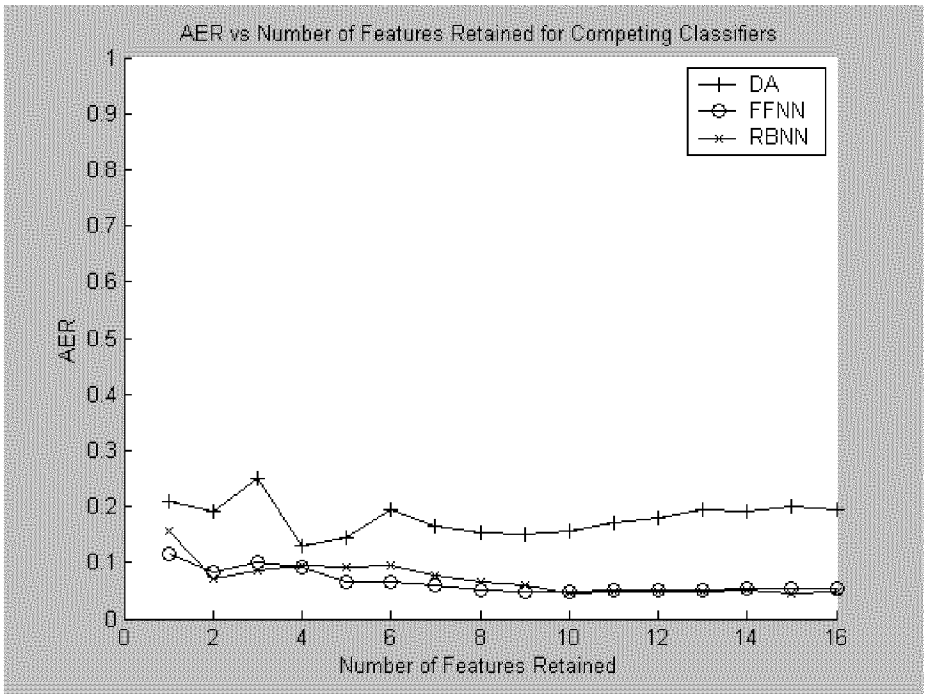

Figure 4-9. AER vs. Number of Features Retained, Experiment 4-5

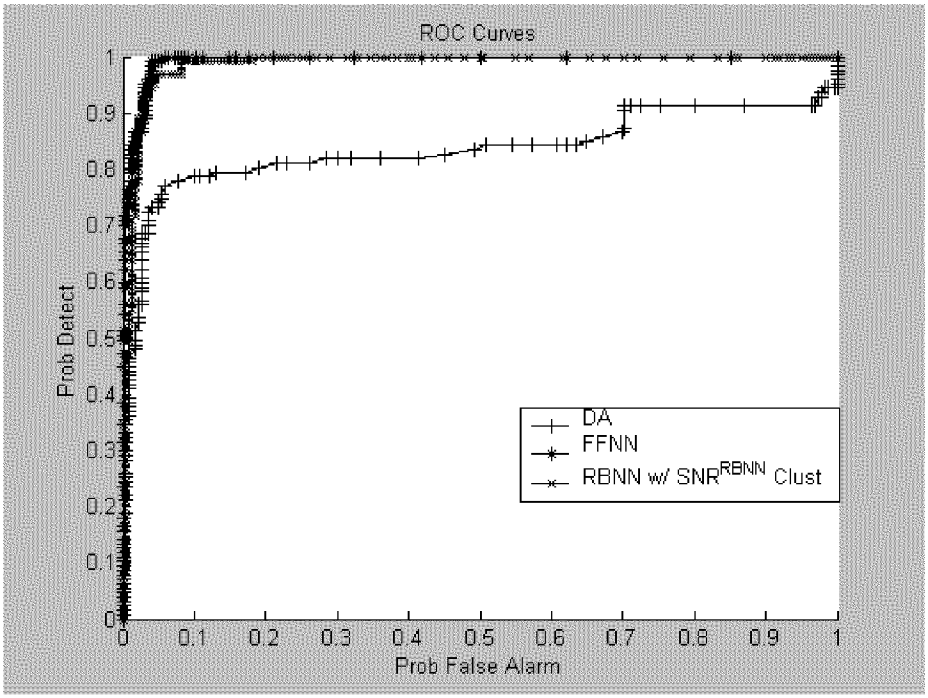

Figure 4-10. ROC Curves for Optimal Stopping Point, Experiment 4-5

4.4 Experiment 4-6: Noise-Corrupted Fisher's Iris Feature Selection Test

Bauer *et. al.* [4] present a noise-corrupted version of Fisher's classic Iris problem. This data consist of 148 exemplars belonging to three classes, with 50 exemplars in Class 1 and 49 each in Class 2 and Class 3. Each exemplar has eight features with the first four features being the original features of sepal length, sepal width, petal length and petal width. The final four features are noise features generated as random permutations of the four real features. Bauer *et. al.* determined that petal width and petal length are the only features required for optimal classification accuracy. The feature selection techniques will be evaluated against these criteria.

4.4.1 Classification for the Three Class Problem

Prior to conducting the experiment, the classification techniques discussed previously must be discussed as they apply to this problem. All of the techniques discussed in Chapter 2 and Chapter 3 are predicated on classification for a two-class problem. Before applying these techniques to a problem with three (or more) classes, some adaptations are required. Only minor changes are required to the actual classifications for the Artificial Neural Networks (ANN) and no changes are necessary to generate the quadratic discriminant scores. The ANN's require three output nodes, instead of the one necessary for the two-class problem. Instead of training the network to one for Class 1 and zero for Class 2, the network changes to the vectors [1,0,0] for exemplars in Class 1, [0,1,0] for Class 2 and [0,0,1] for Class 3. An exemplar is classified in the class corresponding to the node with the largest output.

Most of the differences between the two-class and three-class problems involve feature selection. For DBS, there are now three measures for each exemplar, one for each output differing only in the weight that is applied to the different nodes. The measure now becomes the average of the absolute value of the individual measures

$$ADS_{ik} = \frac{1}{3}\sum_{l=1}^{3}\left|DS^{(l)}{}_{ik}\right| \tag{4.1}$$

where $DS^{(l)}{}_{ik}$ is the saliency measure describe in Equation (3.2) applied to the l^{th} exemplar. Discriminant Loadings require more of an adjustment. Equation (2.11) uses the $\underline{b}$ defined in Equation (2.6) to generate the loadings. This definition of $\underline{b}$ is only valid for two-class problems. Laine [11] recommends estimating $\underline{b}$ for each class

$$\underline{b}_i = \Sigma^{-1}\mu_i \tag{4.2}$$

where Σ is the sample covariance matrix for the whole population and μ_i the sample mean for the i^{th} class. These $\underline{b}_i$ are substituted directly for b in Equation (2.11). The loading for the k^{th} feature becomes the maximum (in absolute value) of the class loadings.

Some of the evaluation methods described in Section 2-5 also need to be adjusted and some of the methods cannot be applied to the three-class problem. Confusion Matrices (CM) and AER are generated in the same manner as for the two-class problem, except that there are nine distinct outcomes rather than four. This difference in composition of the CM prevents the construction of a true ROC Curve, and consequently the mean distance metric is unavailable. The multinomial selection procedure is available however, with only minor changes. The posterior probabilities for DA are calculated by applying Equation (2.38), except that the denominator is now the sum of the three quadratic discriminant scores. The posterior probabilities for the ANN's are even simpler than those described in Section 2-5-3. The posterior probabilities for each class are the outputs (in the case of RBNN's, these outputs are standardized to the interval [0,1]) for the corresponding node of the trained networks. The evaluation of this three-class problem will entail comparison of the feature selection techniques in parsimony and the general classification will be evaluated using AER and the multinomial selection criteria.

4.4.2 Results for the Noise-Corrupted Iris Problem Feature Selection Test

The Fisher's Iris data were divided into 75 exemplars for training and 73 for validation. Fifteen of the training exemplars were allotted for internal validation for the FFNN. Additionally, the FFNN used twelve hidden nodes. The RBNN began with 75 centers which were reduced to three after the first five iterations. The results of this experiment are given in Figure 4-11 and Table 4-10. The optimal stopping point for the RBNN and FFNN was with two features remaining, petal width and petal length, with petal width being the most salient feature. The optimal feature set for DA included these two features plus sepal length. All three feature selection techniques produced similar feature sets and identical estimates of the AER. The optimal FFNN however, significantly outperformed DA and the RBNN in the multinomial selection criteria. This result is consistent with the previous experiments.

Table 4-10. Metrics for Optimal Classifiers, Experiment 4-6

Measures	DA	FFNN	RBNN
Features Retained	3	2	2
Noise Features Retained	0	0	0
AER	0.0137	0.0137	0.0137
90% CI Half-Width	0.0326	0.0326	0.0326
Multinomial	0.0000	0.9863	0.0137
90% CI Half-Width	0.0543	0.0326	0.0326

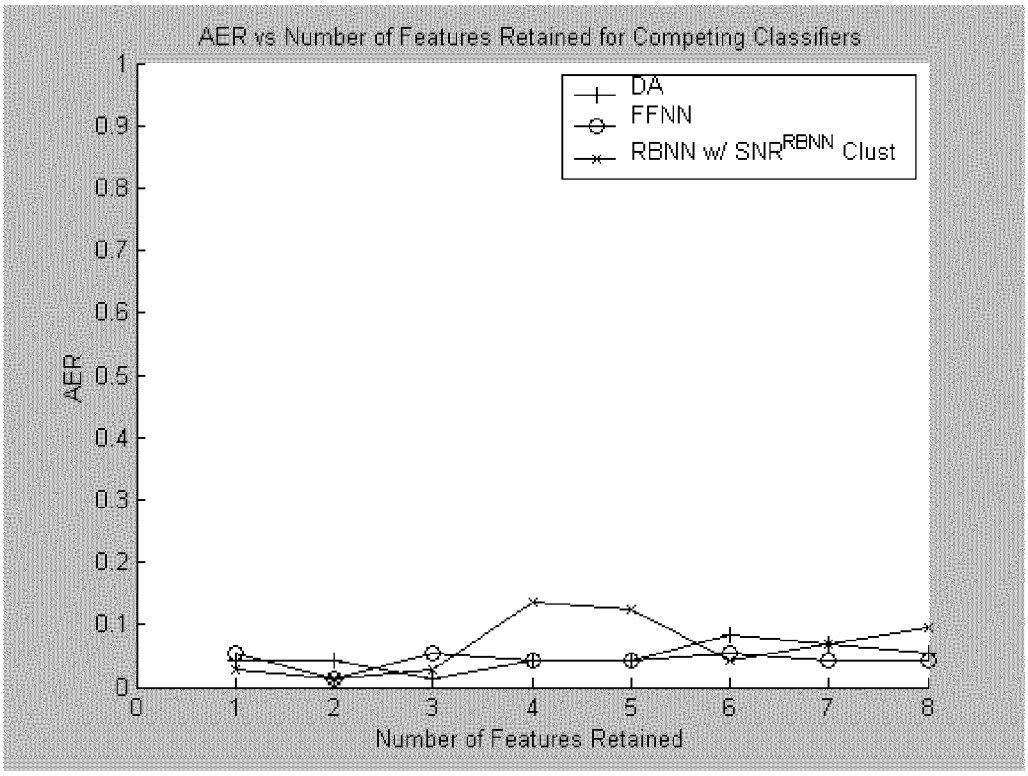

Figure 4-11. AER vs. Number of Features Retained, Experiment 4-6

In this chapter three primary problems were explored: Block-C, the University of Wisconsin Breast Cancer Data set and Fisher's Iris problem. For all problems RBNN's perform at least as well as FFNN's in AER and in the ROC Curves. However, the FFNN's performed consistently better in the mean distance and multinomial selection metrics. For this reason, the FFNN's performed significantly better than the RBNN's when applied to the perturbed data sets. For the two feature selection tests, Experiment 4-5 and Experiment 4-6, the integrated architecture and feature selection algorithm for the RBNN performed as well as Discriminant Loadings and SNR.

5 Summary and Recommendations

5.1 Overview

This chapter will summarize the existing techniques presented, as well as the newly developed algorithms, for solving an integrated architecture design and feature selection problem for radial basis neural networks. Additionally, this chapter will highlight the major contributions of the thesis and give recommendations for significant areas of future research.

5.2 Summary of Techniques

This thesis presented several feature selection techniques including Discriminant Loadings applied to Discriminant Analysis (DA) and signal-to-noise ratio (SNR) applied to Feed Forward Neural Networks (FFNN). Clustering techniques for Radial Basis Neural Networks (RBNN) were also discussed. The two techniques applied to the experiments were *K*-Means and Radial Basis Function Iterative Construction Algorithm (RICA). Chapter 3 developed three additional techniques for RBNN's. The first technique was feature selection using derivative-based saliency (DBS). The second technique was a new clustering algorithm, SNR^{RBNN} used for architecture selection in RBNN's. These techniques were combined to form the integrated architecture and feature selection algorithm which alternates between clustering and feature selection until the appropriate centers and features are retained. Table 5-1 details the techniques and Table 5-2 illustrates to which experiments they were applied.

Four analysis techniques were also discussed in this thesis: Actual Error Rate (AER), visual inspection of the Receiver Operating Characteristic (ROC) Curves, the mean distance metric, and the multinomial selection procedure. These techniques were applied to the experiments to evaluate the competing classifiers.

Table 5-1. Description of Classification Techniques

Technique	Classifier	Application	Description
DL	DA	Feature Selection	Discriminant Loadings – Measures the correlation between the output and features
SNR	FFNN	Feature Selection	Signal-to-Noise Ratio – A weight-based saliency measure contrasting features to a noise feature
DBS	RBNN	Feature Selection	Derivative-Based Saliency – Measures the unit change in the output with respect to the feature
SNR^{RBNN}	RBNN	Architecture Selection	Signal-to-Noise Ratio Clustering– A weight-based clustering algorithm contrasting centers to a noise center
***K*-Means**	RBNN	Architecture Selection	K-Means Clustering Algorithm – A clustering algorithm using Euclidean distance
RICA	RBNN	Architecture Selection	Radial Basis Function Iterative Construction Algorithm – A clustering algorithm using Mahalanobis distance

Table 5-2. Summary of Experiments and Techniques.

Experiment	Data Set	Purpose	Techniques						
			DA	FFNN	RBNN				
			DL	SNR	*K*-Means	RICA	SNR^{RBNN}	DBS	SNR^{RBNN} + DBS
Experiment 3-1	Simple Noise	Feature Selection	X	X	X			X	
Experiment 3-2	Block-C	Clustering			X	X	X		
Experiment 3-3	Simple Noise	Feature Selection	X	X					X
Experiment 4-1	Block-C	Classifier Comparison			X		X		
Experiment 4-2	Perturbed Block-C	Classifier Robustness			X		X		
Experiment 4-3	UW BCD	Classifier Comparison					X		
Experiment 4-4	Perturbed UW BCD	Classifier Robustness					X		
Experiment 4-5	Noisy UW BCD	Feature Selection	X	X					X
Experiment 4-6	Noisy Iris	Feature Selection	X	X					X

5.3 Summary of Contributions

The major contribution of this thesis is an integrated architecture and feature selection algorithm for RBNN's. The performance of this algorithm was comparable to Discriminant Loadings for DA and SNR for FFNN's. It also significantly reduced the number of centers required for optimal classification. Incorporating SNR^{RBNN} for architecture selection and DBS for feature selection provides a viable feature selection routine for RBNN's which is not currently in existence. Additionally, a new clustering algorithm was developed that uses the network to determine the necessary architecture. The new integrated algorithm is suitable for any classification problem. Examples of potential application areas include the classification of failure modes from sensor data on various aircraft components, classifying individuals as pass or fail for pilot training, and discriminating targets from clutter for target recognition systems.

5.4 Conclusions

There are several general conclusions that can be drawn from this research. This thesis highlights the need for feature selection, and illustrates why the development of feature selection for RBNN's is important. Experiment 3-3 illustrated the effect of noise on classification accuracy. For all classifiers considered, the AER is significantly worse for the data with a large number of noise features versus the data with only the true feature. This effect is more pronounced in the absence of strong features. Experiment 3-3 has significant overlap between the two classes with a minimum error rate of approximately 16%. Experiment 4-5 has less inherent error, and Experiment 4-6 has features which will almost perfectly discriminate between the three populations. For these latter two experiments the noise does not negatively impact classification accuracy for the Artificial Neural Networks. The AER for DA is significantly worse for the noise corrupted data in Experiment 4-5, but not nearly as much as in Experiment 3-3. For

Experiment 4-6, the effect of noise on the AER is eliminated. These experiments illustrate the need of feature selection in the absence of strong features, particularly for DA.

This research also highlights the variable performance of the classifiers across the different experiments. FFNN's and RBNN's are consistently the top performers for all the applications. DA, while performing as well as the ANN's in Experiments 4-3 and 4-6, performed significantly worse than the ANN's in all measures for the other experiments. The performance of FFNN's and RBNN's are similar with two important distinctions: 1) RBNN's outperform FFNN's in AER for the geometric Block-C problem of Experiment 4-1, 2) the ROC Curves for the RBNN's dominate the FFNN across the training set sizes. For this problem, the RBNN outperforms the FFNN.

While the RBNN's perform better than the FFNN in AER in Experiment 4-1 and comparably for the other experiments, FFNN's consistently perform better in the mean distance and multinomial selection metrics. The FFNN provides more confidence in the classification results than DA and RBNN's for all the applications in this thesis. The impact of the performance in the mean distance metric is illustrated in Experiments 4-2 and 4-4 where the validation set is perturbed. In both instances, the FFNN's outperform the other two classifiers. Of particular interest is Experiment 4-2 in which the FFNN's outperform the RBNN's for the perturbed data set, while the RBNN's outperform the FFNN's for the standard data. These results indicate a fundamental difference in the problems best suited for the ANN's. RBNN's are better suited for applications where the validation set is distributed identically to the training set and no deviations are expected for new data. FFNN's are more resistant to these deviations and are better suited to applications where the new exemplars might change in time. This is particularly true of problems involving human data that are to be applied in the long run.

5.5 Recommendations for Future Research

The results of this research identify many fruitful areas of future research. Since most of the work performed in this thesis was experimental in nature, it would be instructive to test the algorithm on problems other than the four discussed herein. Through additional experimentation, it may be possible to gain further insight into the performance of the integrated algorithm as compared to existing techniques.

Second, it may be possible to improve upon the procedure for selecting the number and location of the centers. In particular, this may be accomplished by training the centers as in [12]. Implementing this approach, in conjunction with derivative-based saliency, should be more computationally efficient.

Finally, the empirical results provide some insight into theoretical relationships between the signal-to-noise ratio clustering algorithm and the K-means clustering approach. It would be instructive to explore this relationship analytically to determine if, in fact, the ROC curves for the two approaches converge or if this is simply an artifact of the data sets considered.

Appendix A. Derivation of Derivative-Based Saliency for RBNN's

The network output, z, of the i^{th} exemplar is

$$z^{(i)} = \sum_{j=1}^{p} w_j \exp\left[\frac{-1}{2\sigma_j^{\,2}} \sum_{k=1}^{m} \left(x_k^{\,(i)} - \mu_k^{\,(j)}\right)^2\right] \tag{A.1}$$

where w_j is the weight of the j^{th} center, p is the number of nodes, $\mu_k^{(j)}$ is the k^{th} component of the j^{th} center and m is the number of features. This can also be written as

$$z^{(i)} = \sum_{j=1}^{p} w_j \prod_{k=1}^{m} \exp\left[\frac{-1}{2\sigma_j^{\,2}} \left(x_k^{\,(i)} - \mu_k^{\,(j)}\right)^2\right] \tag{A.2}$$

The partial derivative of this output with respect to feature l becomes

$$DS_{il} = \partial \frac{z^{(i)}}{\partial x_l} = \sum_{j=1}^{p} w_j \sum_{k=1}^{m} \frac{\partial}{\partial x_l} \left(\prod_{k=1}^{m} \exp\left[\frac{-1}{2\sigma_k^{\,2}} \left(x_k^{\,(i)} - \mu_k^{\,(j)}\right)^2\right]\right) \tag{A.3}$$

Applying the chain rule, this becomes

$$\partial \frac{z^{(i)}}{\partial x_l} = \sum_{j=1}^{p} w_j \sum_{k=1}^{m} \left\{ \frac{\partial}{\partial x_l} \left(\exp\left[\frac{-1}{2\sigma_j^{\,2}} \left(x_k^{\,(i)} - \mu_k^{\,(j)}\right)^2\right]\right) \left(\prod_{\substack{q=1 \\ q \neq k}}^{m} \exp\left[\frac{-1}{2\sigma_j^{\,2}} \left(x_q^{\,(i)} - \mu_q^{\,(j)}\right)^2\right]\right)\right\} \tag{A.4}$$

For $l \neq k$

$$\frac{\partial}{\partial x_l} \left(\exp\left[\frac{-1}{2\sigma_j^{\,2}} \left(x_k^{\,(i)} - \mu_k^{\,(j)}\right)^2\right]\right) = 0 \tag{A.5}$$

For $l = k$

$$\frac{\partial}{\partial x_l} \left(\exp\left[\frac{-1}{2\sigma_j^{\,2}} \left(x_k^{\,(i)} - \mu_k^{\,(j)}\right)^2\right]\right) = \frac{-1}{\sigma_j^{\,2}} \left(x_k^{\,(i)} - \mu_k^{\,(j)}\right) \exp\left[\frac{-1}{2\sigma_j^{\,2}} \left(x_k^{\,(i)} - \mu_k^{\,(j)}\right)^2\right] \tag{A.6}$$

Therefore, Equation (App.3) becomes the result seen in Equation (3.2)

$$DS_{il} = \partial \frac{z^{(i)}}{\partial x_l} = \sum_{j=1}^{p} \frac{-w_j}{\sigma_j^{\,2}} \left(x_l^{\,(i)} - \mu_l^{\,(j)}\right) \prod_{k=1}^{m} \exp\left[\frac{-1}{2\sigma_j^{\,2}} \left(x_k^{\,(i)} - \mu_k^{\,(j)}\right)^2\right] \tag{A.7}$$

Appendix B. Derivation of Derivative-Based Saliency for GRNN's

The DBS measures for GRNN's are obtained in a similar fashion to RBNN's. The network output for the i^{th} exemplar is

$$\frac{z^{(i)}}{s^{(i)}} = \frac{\sum_{j=1}^{p} w_j \exp\left[\frac{-1}{2\sigma_j^{\,2}} \sum_{k=1}^{m} \left(x_k^{\,(i)} - \mu_k^{\,(j)}\right)^2\right]}{\sum_{j=1}^{p} \exp\left[\frac{-1}{2\sigma_j^{\,2}} \sum_{k=1}^{m} \left(x_k^{\,(i)} - \mu_k^{\,(j)}\right)^2\right]} \tag{B.1}$$

This is the sum of the weighted hidden outputs, $z^{(i)}$ scaled by the unweighted hidden outputs, $s^{(i)}$. The partial derivatives of this expression with respect to the l^{th} feature is obtained by using the quotient rule, and is given by

$$DS_{il} = \frac{\partial}{\partial x_l} \frac{z^{(i)}}{s^{(i)}} = \frac{s^{(i)} \frac{\partial}{\partial x_l} z^{(i)} - z^{(i)} \frac{\partial}{\partial x_l} s^{(i)}}{s^{(i)2}} \tag{B.2}$$

The partial derivative of $z^{(i)}$ is given in Equation (A.7), with the partial derivative of $s^{(i)}$ differing only in the absence of the weights. Therefore, the saliency measure is

$$DS_{il} = \frac{s^{(i)} \sum_{j=1}^{m} \frac{-w_j}{\sigma_j^{\,2}} \left(x_l^{\,(i)} - \mu_l^{\,(i)}\right) h_{ij} - z^{(i)} \sum_{j=1}^{m} \frac{-1}{\sigma_j^{\,2}} \left(x_l^{\,(i)} - \mu_l^{\,(i)}\right) h_{ij}}{s^{(i)2}} \tag{B.3}$$

where

$$h_{ij} = \exp\left[\frac{-1}{2\sigma_j^{\,2}} \left(\underline{x}^{(i)} - \underline{\mu}^{(j)}\right)^T \left(\underline{x}^{(i)} - \underline{\mu}^{(j)}\right)\right] \tag{B.4}$$

Bibliography

1. Alsing, S. G. *The Evaluation of Competing Classifiers*. Air Force Institute of Technology (AU), Wright Patterson AFB OH, Mar 2000 (AD-A375294).

2. Backer, Eric. *Computer-assisted Reasoning in Cluster Analysis,* New York: Prentice Hall, 1995.

3. Bauer, Kenneth L. Class Notes, OPER 685, Multivariate Data Analysis, Graduate School of Engineering and Management, Air Force Institute of Technology, Wright Patterson AFB OH, Spring Quarter 2001.

4. Bauer, Kenneth L, Stephen G. Alsing, Kelly A. Green. "Feature Screening Using Signal-to-Noise Ratios," *Neurocomputing, 31*: 29-44 (2000).

5. D'Agostino, Ralph and Michael A. Stephens, editors. *Goodness-Of-Fit Techniques,* New York: Marcel Dekker, Inc, 1986.

6. Dillon, William R. and Matthew Goldstein. *Multivariate Analysis Methods and Applications,* New York: John Wiley and Sons, 1984.

7. Foor, Wesley E. Adaptive Optical Radial Basis Function Neural Network Classifier. Tech. rep., Rome Laboratory, Griffiss AFB NY, Dec 1994.

8. Karson, Marvin J. *Multivariate Statistical Methods,* Ames Iowa: The Iowa State University Press, 1982.

9. Kohlmorgen, J, S. Lemm., G. Ratch, and K. Muller. "Analysis of Nonstationary Time Series by Mixtures of Self-Organizing Predictors," *Proceedings of the 2000 IEEE Processing Society Workshop, Volume 1.* 85-94. New York: IEEE Press, 2000.

10. Krishnaiah, P.R. *Handbook of Statistics: v.1,* Amsterdam, The Netherlands: Elsevier Science Publisher B.V., 1988.

11. Laine, Trevor I. *Selection of Psychophysiological Features Across Subjects for Classifying Workload Using Artificial Neural Networks.* MS Thesis, AFIT/GOR/ENS/99M-09. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 1999 (AD-A361613).

12. Looney, Carl G. *Pattern Recognition Using Neural Networks,* New York: Oxford University Press, 1997.

13. Malkovich, J. F. and A. A. Afifi. "On Tests for Multivariate Normality," *Journal of the American Statistical Association,* 68: 176-179 (March 1973),

14. Shapiro, S. S. and M. B. Wilk. "An Analysis of Variance Test for Normality," *Biometrika* 52: 591-611 (1965).

15. Silverman, B. W. *Density Estimation for Statistics and Data Analysis,* London: Chapman and Hall, 1986.

16. Steppe, J. M., K. W. Bauer, and S. K. Rogers. "Integrated Feature and Architecture Selection". *IEEE Transactions on Neural Networks*, Vol. 7, 4: 1007-1014 (July 1996).

17. Trunk, G. V. "A Problem of Dimensionality: A Simple Example," *IEEE Transactions Pattern Analysis and Machine Intelligence* 1: 306-307 (July 1975).

18. University of California-Irvine (UCI). *Machine Learning Repository.* http://www.ics.uci.edu/mlearn/, 1999.

19. Wasserman, Philip D. *Advanced Methods in Neural Computing,* New York: Van Nostrand Reinhold, 1993.

20. Weigend, Andreas S. and Neil A. Geshenfeld. *Time Series Prediction,* Reading Massachusetts: Perseus Books Publishing, L.L.C., 1994.

21. Wilson, Terry A.. *Automatic Target Cueing of Hyperspectral Image Data.* Air Force Institute of Technology (AU), Wright Patterson AFB OH, Sep 1998 (AD-A289429).

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to an penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From – To)*
01-09-2002	Master's Thesis	Jun 2001 – Sep 2002

4. TITLE AND SUBTITLE

AN INTEGRATED ARCHITECTURE AND FEATURE SELECTION ALGORITHM FOR RADIAL BASIS NEURAL NETWORK

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Flietstra, Timothy D.., Captain, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 P Street, Building 640, WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GOR/ENS/02-07

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

N/A

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The research contribution of this thesis is the first known integrated architecture and feature selection algorithm for Radial Basis Neural Networks (RBNN's). The objective is to apply the network iteratively to determine the final architecture and feature set used to evaluate a problem. Additionally, this thesis compares three different classification techniques, Discriminant Analysis (DA), Feed-Forward Neural Networks (FFN) and RBNN's against several hard to solve problems. These problems were used to evaluate general classifier performance as well as the performance of the feature selection techniques.

This thesis describes the classification techniques as well as the measures used to evaluate them. It next develops a new clustering technique used to determine the network architecture and the saliency measure used to select features for RBNN's. Next, the thesis applies these techniques to three general problems, Block-C, the University of Wisconsin Breast Cancer Data (UWBCD) and a noise corrupted version of Fisher's Iris problem. Finally, the conclusions and recommendations for future research are provided.

15. SUBJECT TERMS

Discriminant Analysis, Artificial Neural Networks, Feed-Forward Neural Networks, Radial Basis Neural Networks
Feature Selection, Discriminant Loadings, Weight-Based Saliency, Derivative-Based Saliency
Receiver Operating Characteristic Curves, Confusion Matrices

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Kenneth W. Bauer, Ph. D. (ENS)
U	U	U	UU	82	19b. TELEPHONE NUMBER *(Include area code)* (937) 255-6565, ext 4328; e-mail: Kenneth.Bauer@afit.edu

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18